Machine Learning & Data Mining

Lecture 8

Corina Besliu

Technical University of Moldova

December 3, 2021

Outline

- Types of Datasets
- Predictions with time series
- Modeling time series
- AR, MA, ARMA, ARIMA, SARIMA, SARIMAX, VARMA
- Time series transformations

Outline

- Types of Datasets
- Predictions with time series
- Modeling time series
- AR, MA, ARMA, ARIMA, SARIMA, SARIMAX, VARMA
- Time series transformations

Types of Data: Cross-Sectional Data

Cross-Sectional Data

- Multiple individuals ($i = n$)
- At one point in time ($t = 1$)

Types of Data: Cross-Sectional Data

Cross-Sectional Data

- Multiple individuals ($i = n$)
- At one point in time ($t = 1$)

Types of Data: Cross-Sectional Data

Cross-Sectional Data

- Multiple individuals ($i = n$)
- At one point in time ($t = 1$)

Sartup	Profit	Marketing	R&D
1	150000	147000	73500
2	160000	157000	78500
3	145000	142000	71000
4	121000	118000	59000
5	170000	167000	83500
6	180000	177000	88500
7	190000	187000	93500
8	120000	117000	58500
9	190000	187000	93500
10	240000	237000	118500

Types of Data: Time Series Data

Time Series Data

- One individual ($i = 1$)
- Multiple points in time ($t = n$)

Types of Data: Time Series Data

Time Series Data

- One individual ($i = 1$)
- Multiple points in time ($t = n$)

Types of Data: Time Series Data

Time Series

- One individual ($i = 1$)
- Multiple points in time ($t = n$)

Time	Profit	Marketing	R&D
1.1.2020	150000	147000	73500
2.1.2020	160000	157000	78500
3.1.2020	145000	142000	71000
4.1.2020	121000	118000	59000
5.1.2020	170000	167000	83500
6.1.2020	180000	177000	88500
7.1.2020	190000	187000	93500
8.1.2020	120000	117000	58500
9.1.2020	210000	207000	103500
10.1.2020	180000	177000	88500

Exogenous Regressors are optional

Types of Data: Panel (or Longitudinal) Data

Panel Data

- Multiple individuals ($i = k$)
- Multiple points in time ($t = m$)
- Number of observations: $n = k * m$

Types of Data: Panel (or Longitudinal) Data

Panel Data

- Multiple individuals ($i = k$)
- Multiple points in time ($t = m$)
- Number of observations: $n = k * m$

Types of Data: Panel (or Longitudinal) Data

Panel Data

- Multiple individuals ($i = k$)
- Multiple points in time ($t = m$)
- Number of observations: $n = k * m$

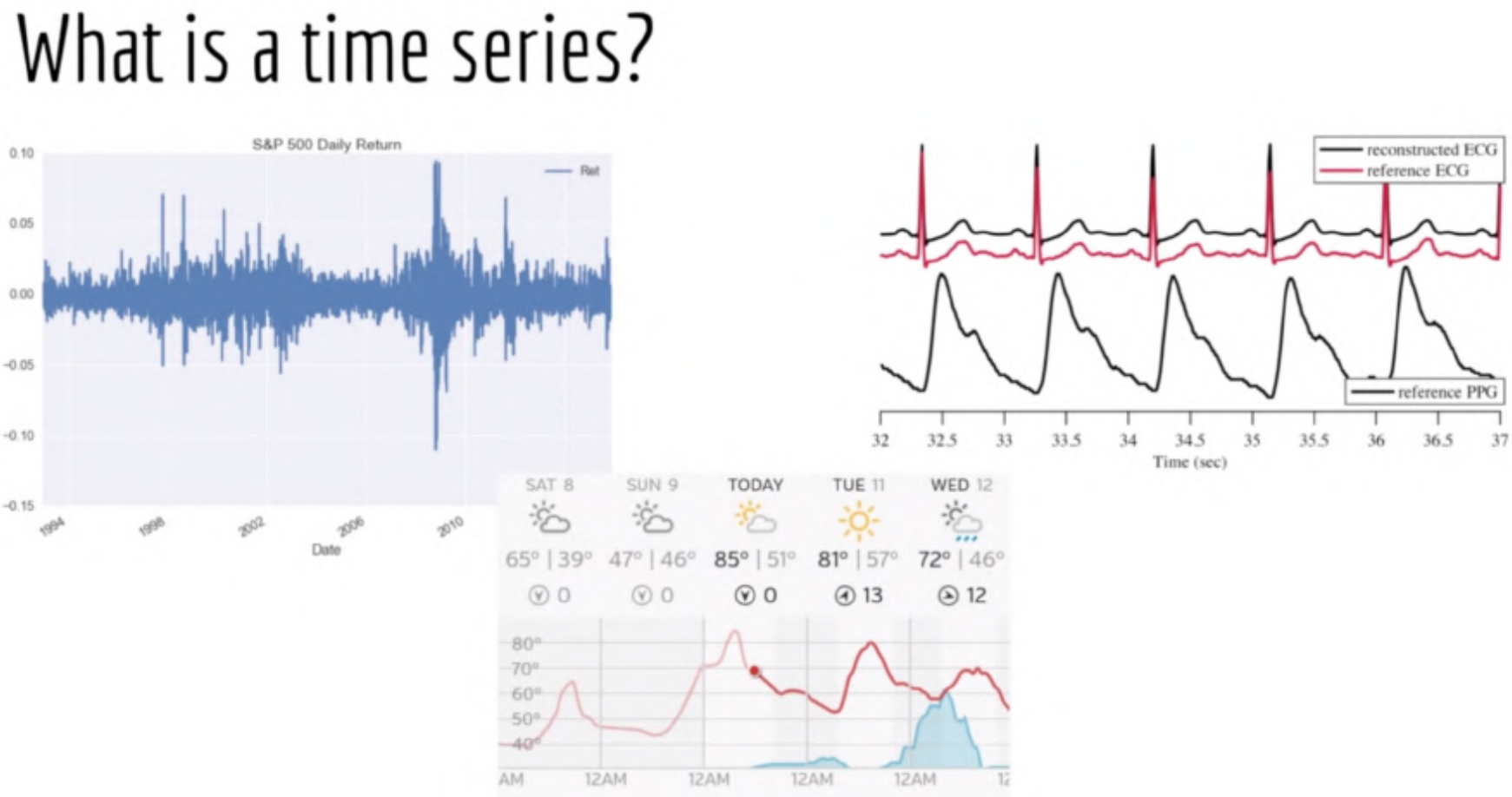
Types of Data: Panel (or Longitudinal) Data

Panel Data

- Multiple individuals ($i = k$)
- Multiple points in time ($t = m$)
- Number of observations: $n = k * m$

Sartup	Time	Profit	Marketing	R&D
1	1.1.2020	150000	147000	73500
1	2.1.2020	160000	157000	78500
1	3.1.2020	145000	142000	71000
1	4.1.2020	121000	118000	59000
1	5.1.2020	170000	167000	83500
1	6.1.2020	180000	177000	88500
1	7.1.2020	190000	187000	93500
1	8.1.2020	120000	117000	58500
1	9.1.2020	190000	187000	93500
1	10.1.2020	240000	237000	118500
2	1.1.2020	150000	147000	144000
2	2.1.2020	160000	157000	154000
2	3.1.2020	145000	142000	139000
2	4.1.2020	121000	118000	115000
2	5.1.2020	170000	167000	164000
2	6.1.2020	180000	177000	174000
2	7.1.2020	190000	187000	184000
2	8.1.2020	120000	117000	114000
2	9.1.2020	190000	187000	184000
2	10.1.2020	240000	237000	234000
3	1.1.2020	150000	147[illegible]0	144000
3	2.1.2020	160000	157000	154000

Examples of Time Series



Specifics of Time Series

Real-world time series are often governed by a (deterministic) trend and they might have (deterministic) cyclical or seasonal components in addition to the irregular/remainder (stationary process) component.

Specifics of Time Series

Real-world time series are often governed by a (deterministic) trend and they might have (deterministic) cyclical or seasonal components in addition to the irregular/remainder (stationary process) component.

Trend component - a long-term increase or decrease in the data which might not be linear. Sometimes the trend might change direction as time increases.

Specifics of Time Series

Real-world time series are often governed by a (deterministic) trend and they might have (deterministic) cyclical or seasonal components in addition to the irregular/remainder (stationary process) component.

Trend component - a long-term increase or decrease in the data which might not be linear. Sometimes the trend might change direction as time increases.

Cyclical component - exists when data exhibit rises and falls that are not of fixed period. The average length of cycles is longer than the length of a seasonal pattern. In practice, the trend component is assumed to include also the cyclical component. Sometimes the trend and cyclical components together are called as trend-cycle.

Specifics of Time Series

Real-world time series are often governed by a (deterministic) trend and they might have (deterministic) cyclical or seasonal components in addition to the irregular/remainder (stationary process) component.

Trend component - a long-term increase or decrease in the data which might not be linear. Sometimes the trend might change direction as time increases.

Cyclical component - exists when data exhibit rises and falls that are not of fixed period. The average length of cycles is longer than the length of a seasonal pattern. In practice, the trend component is assumed to include also the cyclical component. Sometimes the trend and cyclical components together are called as trend-cycle.

Seasonal component - exists when a series exhibits regular fluctuations based on the season (e.g. every month/quarter/year). Seasonality is always of a fixed and known period.

Specifics of Time Series

Real-world time series are often governed by a (deterministic) trend and they might have (deterministic) cyclical or seasonal components in addition to the irregular/remainder (stationary process) component.

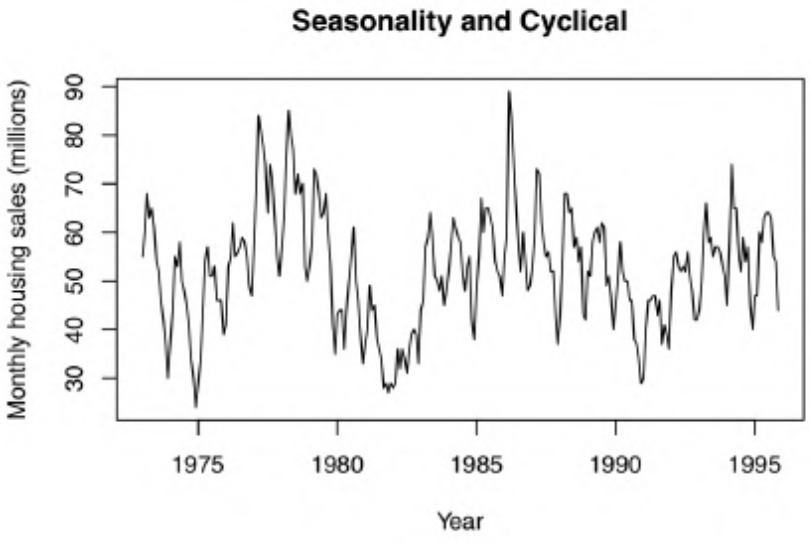
Trend component - a long-term increase or decrease in the data which might not be linear. Sometimes the trend might change direction as time increases.

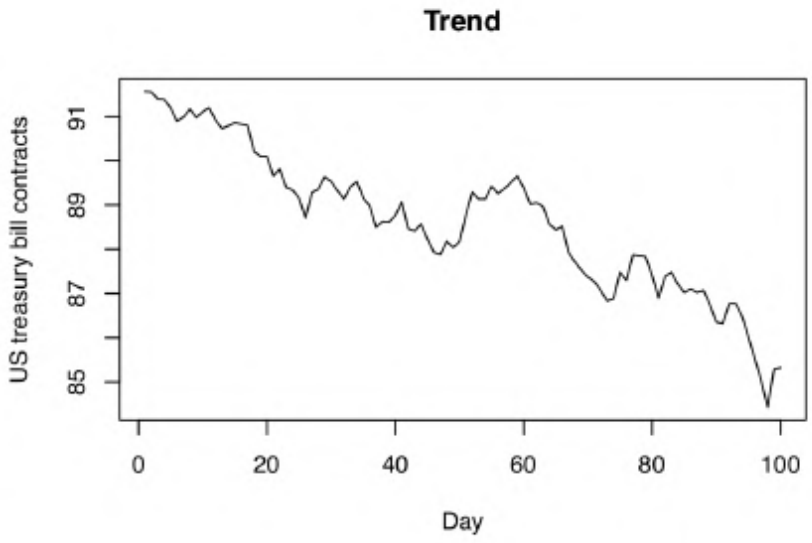
Cyclical component - exists when data exhibit rises and falls that are not of fixed period. The average length of cycles is longer than the length of a seasonal pattern. In practice, the trend component is assumed to include also the cyclical component. Sometimes the trend and cyclical components together are called as trend-cycle.

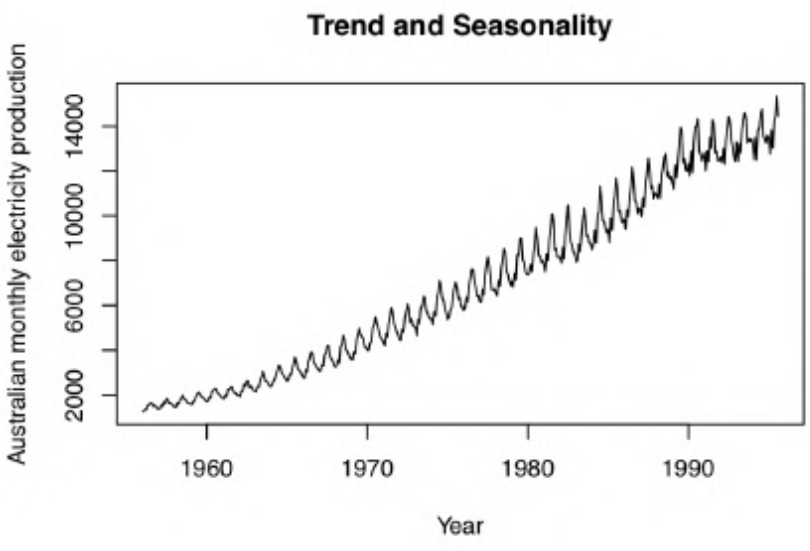
Seasonal component - exists when a series exhibits regular fluctuations based on the season (e.g. every month/quarter/year). Seasonality is always of a fixed and known period.

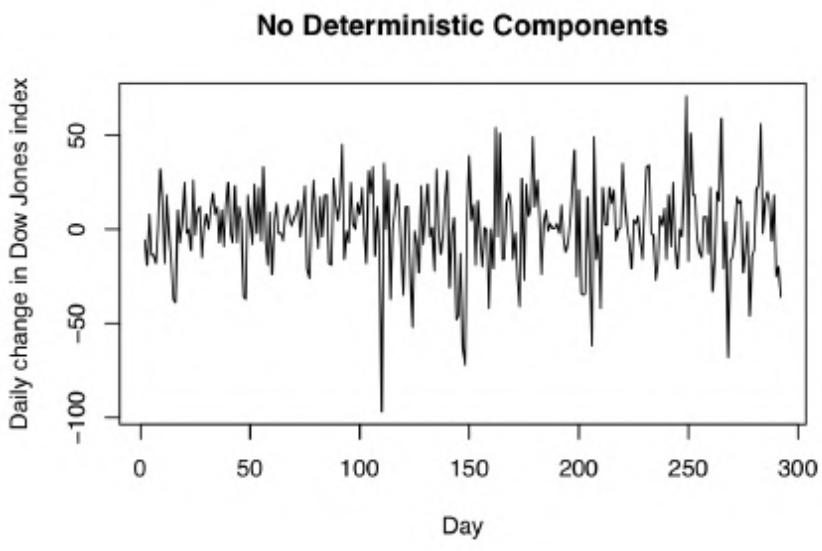
Irregular component - a stationary process.

Specifics of Time Series





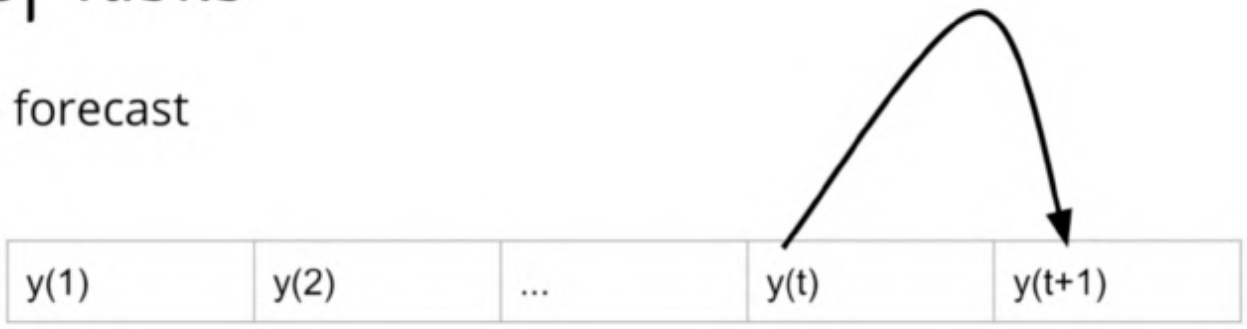




Predicting with Time Series

Types of Tasks

- 1-step forecast



Multi-Step Forecast

- Forecast horizon = number of steps to forecast

Multi-Step Forecast

- Forecast horizon = number of steps to forecast
- E.g. daily sales next week, daily temperature

Multi-Step Forecast

- Forecast horizon = number of steps to forecast
- E.g. daily sales next week, daily temperature
- Imagine a weather channel that only shows 1 day ahead!

Multi-Step Forecast

- Forecast horizon = number of steps to forecast
- E.g. daily sales next week, daily temperature
- Imagine a weather channel that only shows 1 day ahead!
- 1-step forecasts are “not not useful” - use context

Multi-Step Forecast

- Forecast horizon = number of steps to forecast
- E.g. daily sales next week, daily temperature
- Imagine a weather channel that only shows 1 day ahead!
- 1-step forecasts are “not not useful” - use context
- E.g. a brick and mortar shop might forecast the sales of a product next month (from monthly data) to purchase inventory and fulfill demand

Multi-Step Forecast

- Forecast horizon = number of steps to forecast
- E.g. daily sales next week, daily temperature
- Imagine a weather channel that only shows 1 day ahead!
- 1-step forecasts are “not not useful” - use context
- E.g. a brick and mortar shop might forecast the sales of a product next month (from monthly data) to purchase inventory and fulfill demand
- When in doubt: ask your manager / client / stakeholders!

2 ways to produce multi-step forecasts

1) Incremental method (can be done with any 1-step predictor)

2) Multi-output forecast (limited to certain models)

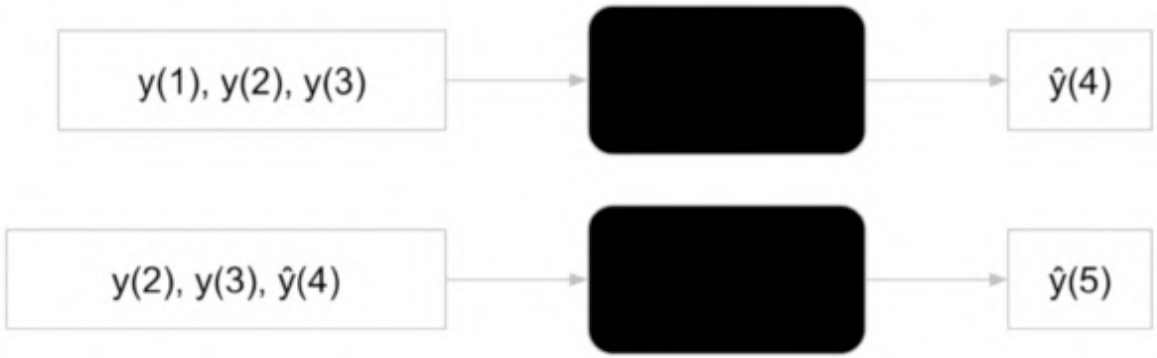
E.g. ARIMA can only do 1-step forecasts, not multi-output forecasts

Example

Let p=3, h=3

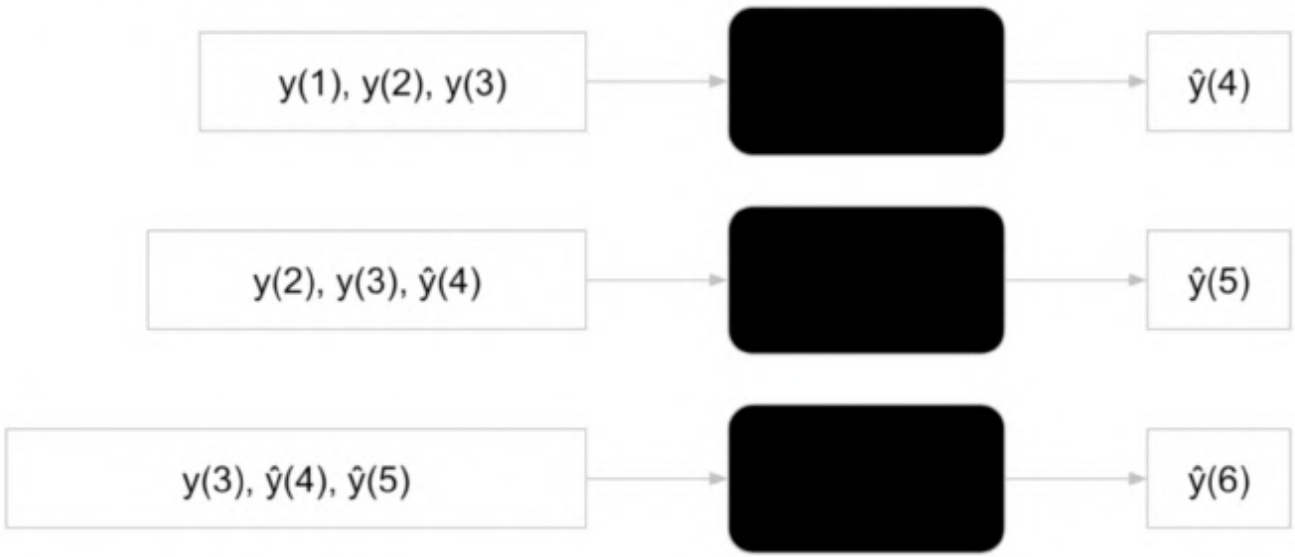
Example

Let p=3, h=3

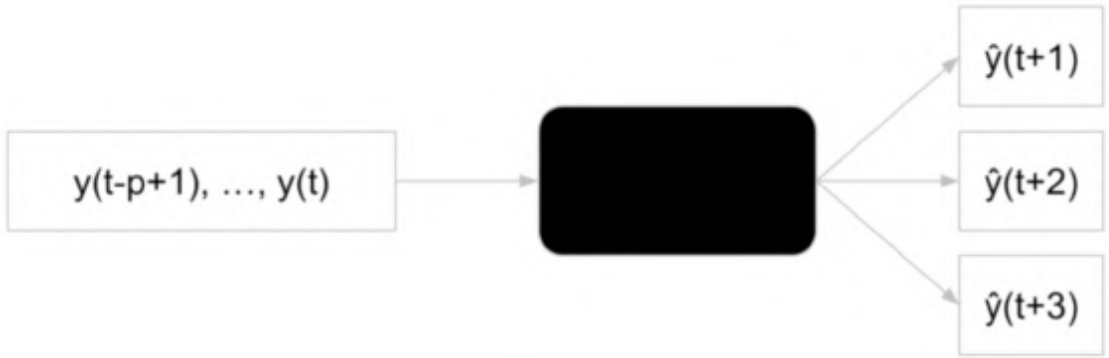


Example

Let p=3, h=3

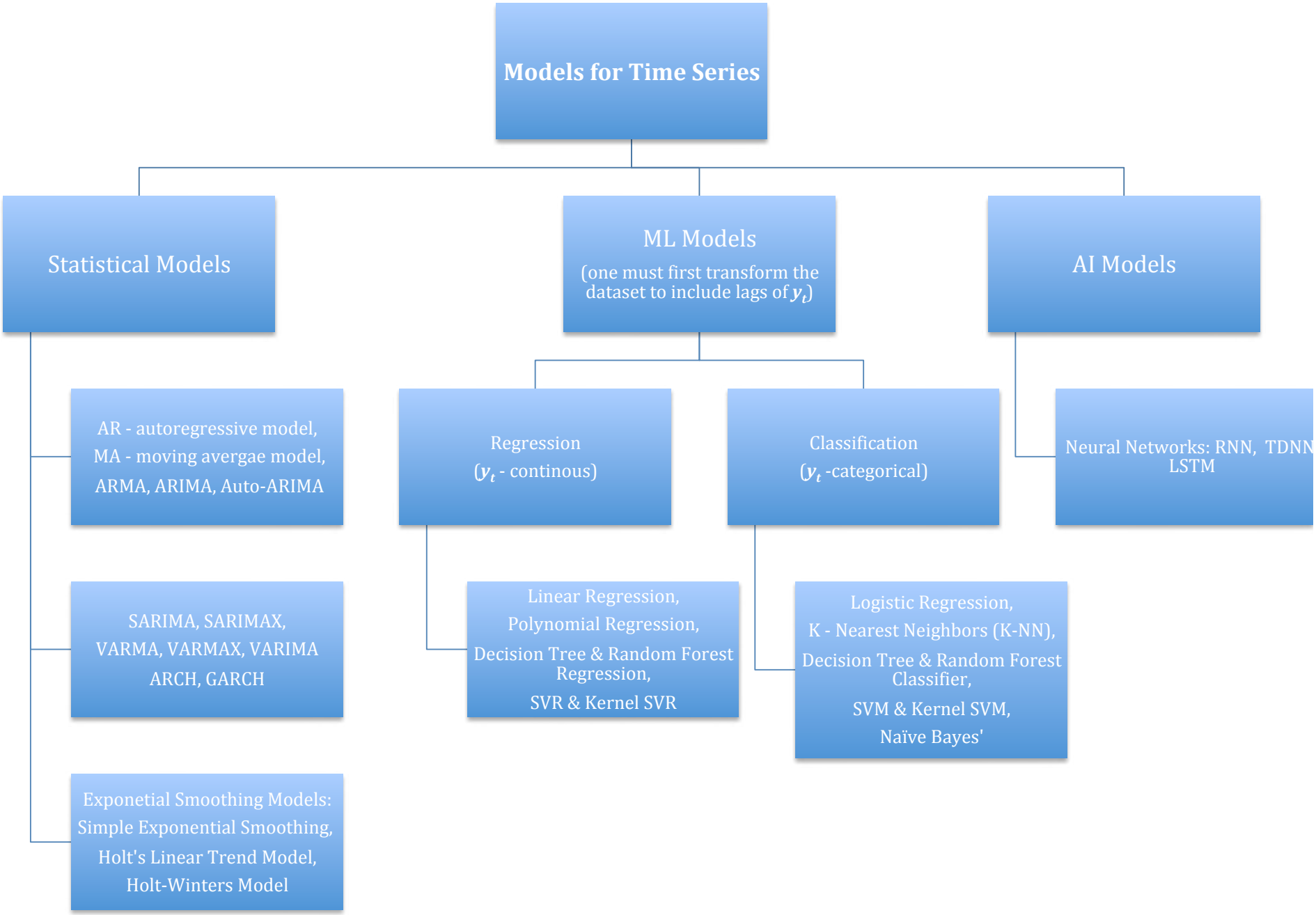


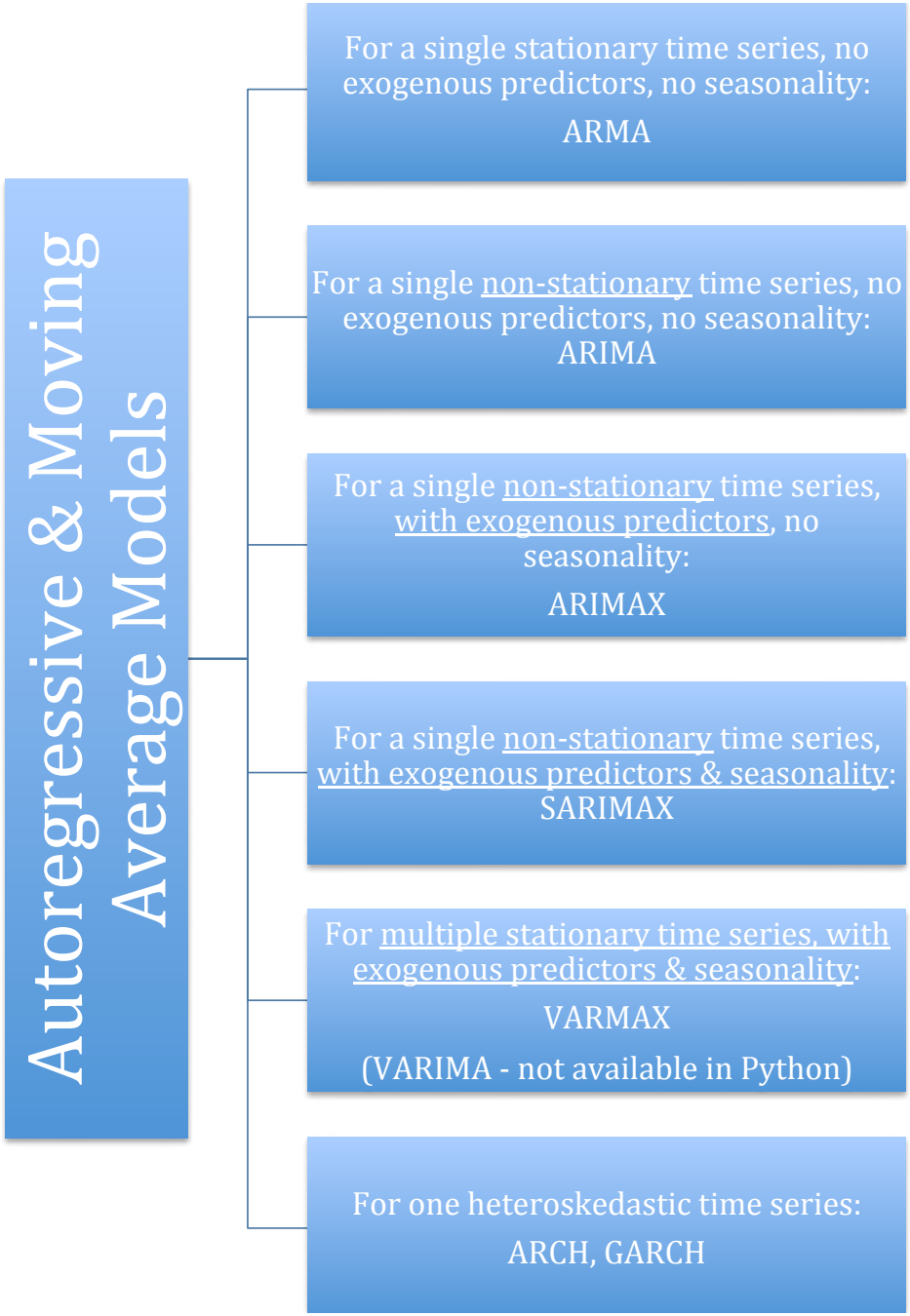
Multi-Output Multi-Step Forecast



Some ML models and deep neural networks can do this

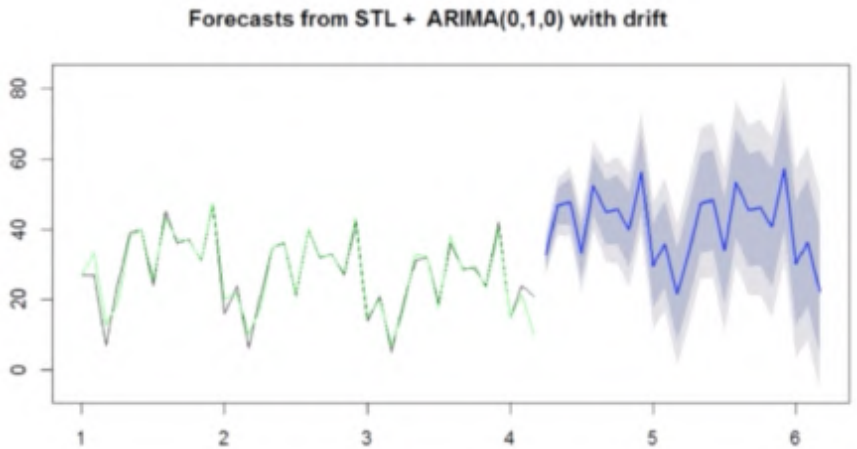
Modeling Time Series


Models for Time Series
Statistical Models
ML Models
(one must first transform the dataset to include lags of y_t)
AI Models
AR - autoregressive model,
MA - moving avergae model,
ARMA, ARIMA, Auto-ARIMA
SARIMA, SARIMAX,
VARMA, VARMAX, VARIMA
ARCH, GARCH
Exponetial Smoothing Models:
Simple Exponential Smoothing,
Holt's Linear Trend Model,
Holt-Winters Model
Regression
(y_t - continous)
Classification
(y_t -categorical)
Neural Networks: RNN, TDNN
LSTM
Linear Regression,
Polynomial Regression,
Decision Tree & Random Forest Regression,
SVR & Kernel SVR
Logistic Regression,
K - Nearest Neighbors (K-NN),
Decision Tree & Random Forest Classifier,
SVM & Kernel SVM,
Naïve Bayes'


Autoregressive & Moving Average Models
For a single stationary time series, no exogenous predictors, no seasonality:
ARMA
For a single non-stationary time series, no exogenous predictors, no seasonality:
ARIMA
For a single non-stationary time series, with exogenous predictors, no seasonality:
ARIMAX
For a single non-stationary time series, with exogenous predictors & seasonality:
SARIMAX
For multiple stationary time series, with exogenous predictors & seasonality:
VARMAX
(VARIMA - not available in Python)
For one heteroskedastic time series:
ARCH, GARCH

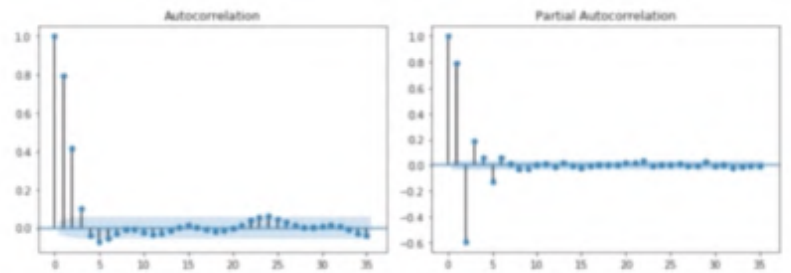
ARIMA Section Introduction

- What is AR(p)?
- What is MA(q)?
- What is ARIMA(p,d,q)?



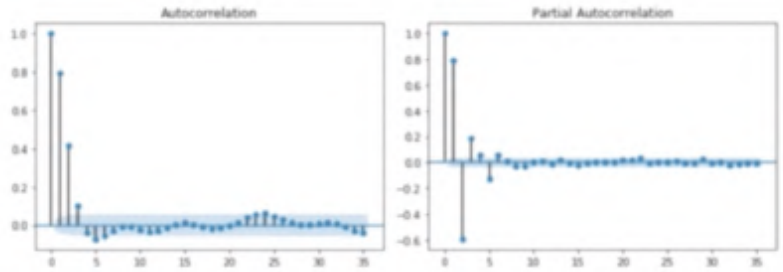
Details and Applications

- How to choose p, d, q?



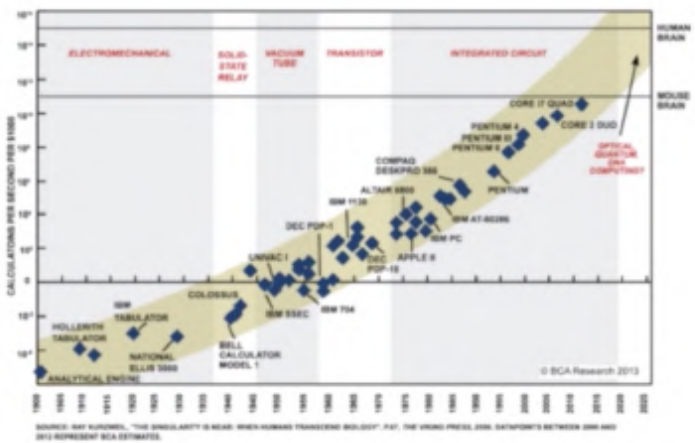
Details and Applications

- How to choose p, d, q?
- ACF and PACF - helps you choose p and q



Auto ARIMA

- We can also just use modern computational power to select p, d, q



Auto ARIMA

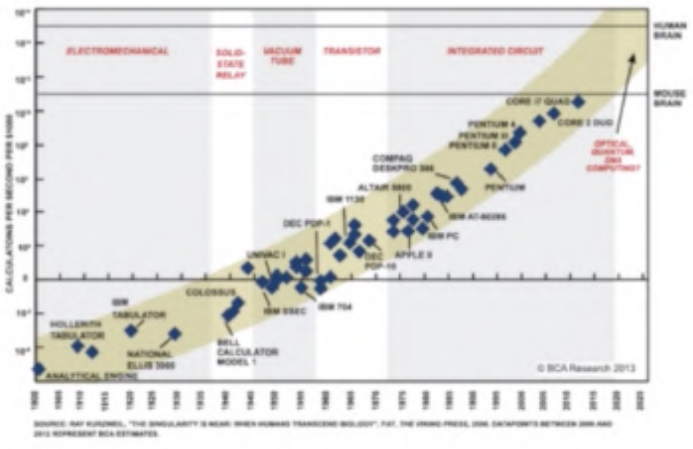
- We can also just use modern computational power to select p, d, q
- Seasonality (SARIMA) and exogenous data (SARIMAX)

Auto ARIMA

- We can also just use modern computational power to select p, d, q
- Seasonality (SARIMA) and exogenous data (SARIMAX)
- What criteria does Auto ARIMA use to define “best”?

Auto ARIMA

- We can also just use modern computational power to select p, d, q
- Seasonality (SARIMA) and exogenous data (SARIMAX)
- What criteria does Auto ARIMA use to define “best”?
- AIC and BIC

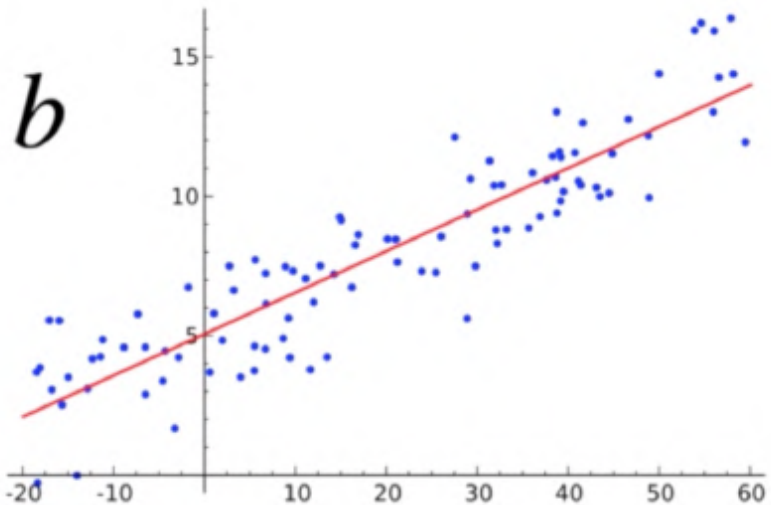


Let's break it down: the AR(p)*IMA* model

What is an autoregressive model?

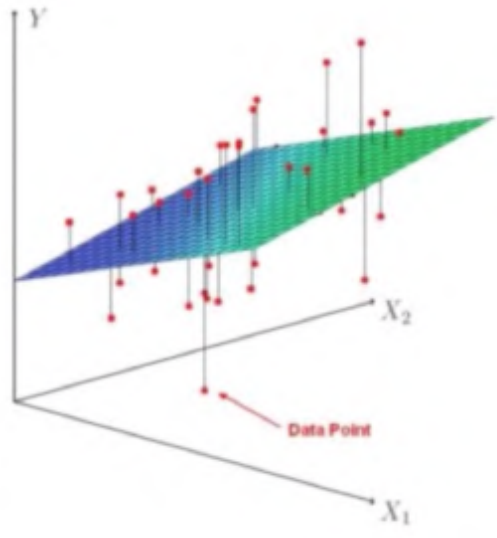
- Let's first recall linear regression
- Example: x = years of experience, y = salary

$$\hat{y} = mx + b$$



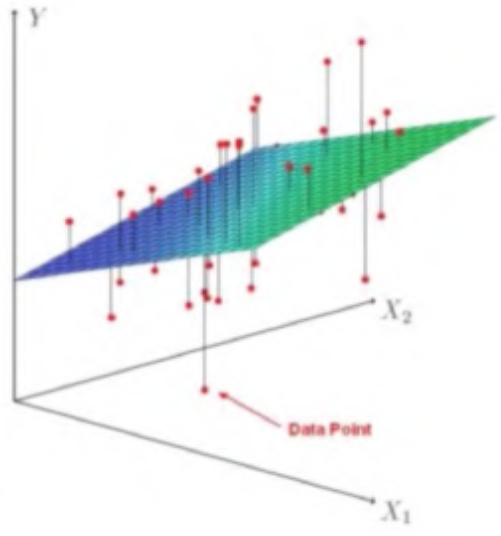
More than one input

$$\hat{y} = w_1 x_1 + w_2 x_2 + b$$



More than one input

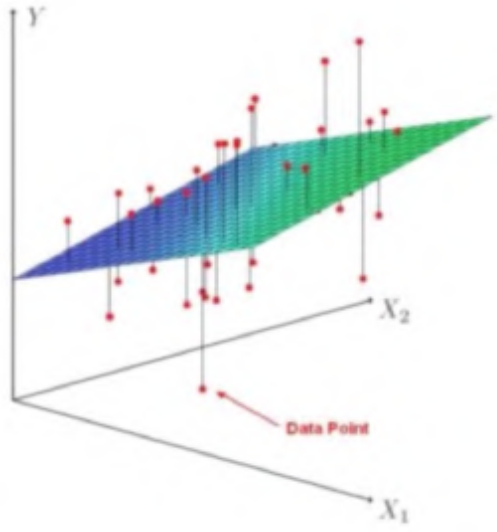
- Ex. x_1 = years of experience, x_2 = age, y = salary



$$\hat{y} = w_1 x_1 + w_2 x_2 + b$$

More than one input

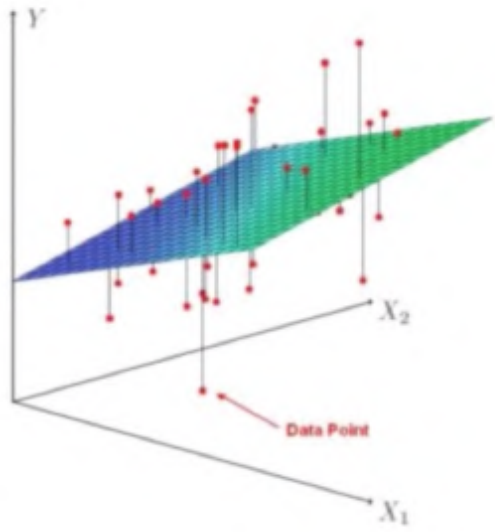
- Ex. x_1 = years of experience, x_2 = age, y = salary
- w_1, w_2, b are found by minimizing the error of the predictions



$$\hat{y} = w_1 x_1 + w_2 x_2 + b$$

More than one input

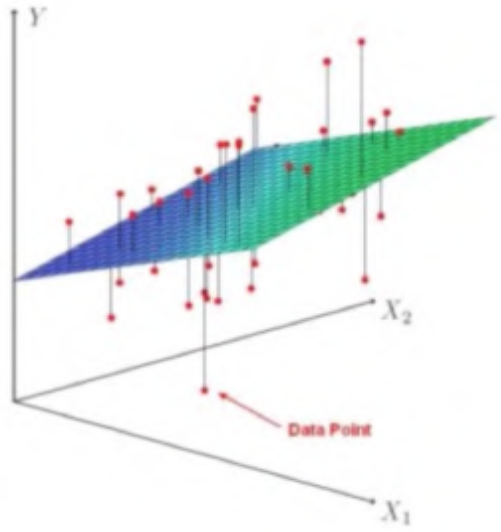
- Ex. x_1 = years of experience, x_2 = age, y = salary
- w_1, w_2, b are found by minimizing the error of the predictions
- b = salary when years of experience and age are zero



$$\hat{y} = w_1 x_1 + w_2 x_2 + b$$

More than one input

- Ex. x_1 = years of experience, x_2 = age, y = salary
- w_1, w_2, b are found by minimizing the error of the predictions
- b = salary when years of experience and age are zero
- w_1 = amount salary increases when years of experience increases by 1



$$\hat{y} = w_1 x_1 + w_2 x_2 + b$$

AR(p)

- It's just linear regression:
  we use the past data points in the series to predict the next point
- AR(p) = autoregressive model of order p

$$\hat{y}_t = b + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \ldots + \varphi_p y_{t-p}$$

Let's do the same thing, but for time series

y1	y2	y3
y2	y3	y4
y3	y4	y5
y4	y5	y6
y5	y6	y7
y6	y7	y8
y7	y8	y9

This is our "X"

y4
y5
y6
y7
y8
y9
y10

This is our "Y"

Let's do the same thing, but for time series

- Can still do model = LinearRegression(), model.fit(X, Y), model.predict(X), model.score(X, Y), etc.

y1	y2	y3
y2	y3	y4
y3	y4	y5
y4	y5	y6
y5	y6	y7
y6	y7	y8
y7	y8	y9

This is our "X"

y4
y5
y6
y7
y8
y9
y10

This is our "Y"

... the *ARI*MA(q) model

MA(q)

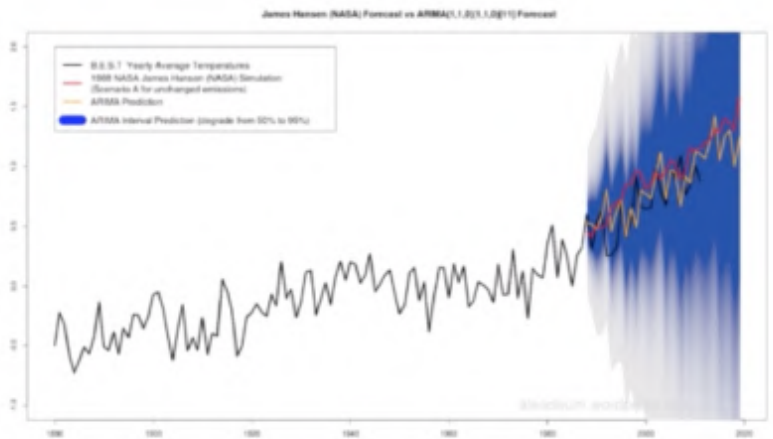
- Moving average of order q
- It depends on past errors, not on past data in the time series

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q}$$

... finally the ARIMA(p,d,q) model

ARIMA

- First, we will discuss ARMA
- Then, we will discuss the "I" component

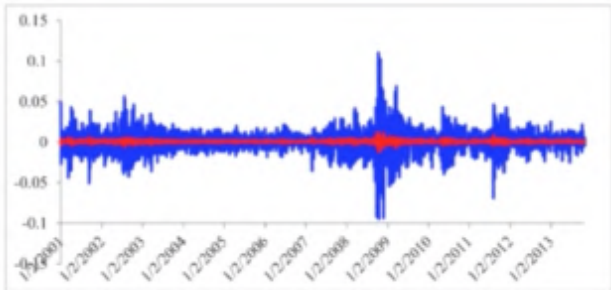


ARMA(p, q)

- ARMA(p, q) = AR(p) + MA(q)

$$y_t = b + \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Differencing

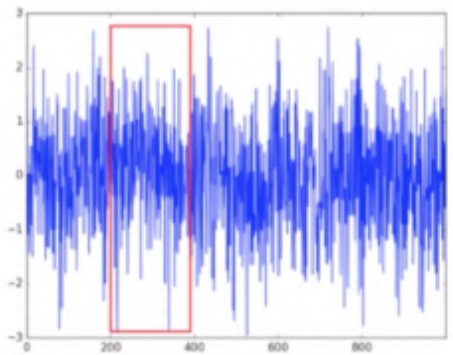


$$Given: \ \{y_t\} = \{y_1, y_2, ..., y_T\} \ (some \ time \ series)$$

$$Differenced \ Series: \ \Delta y_t = y_t - y_{t-1}$$

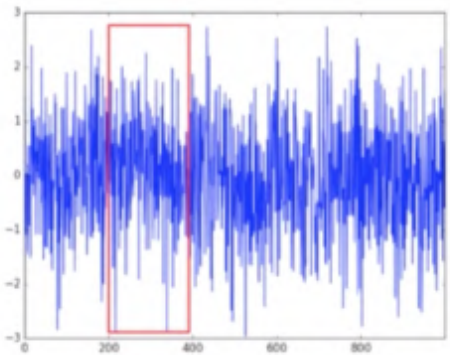
Why difference?

- When fitting an ARMA model, we want the data to be close to stationary



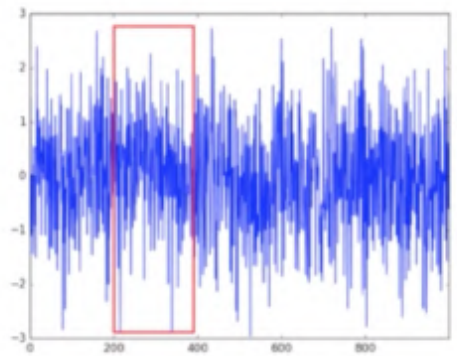
Why difference?

- When fitting an ARMA model, we want the data to be close to stationary
- Stationary = Does not change over time

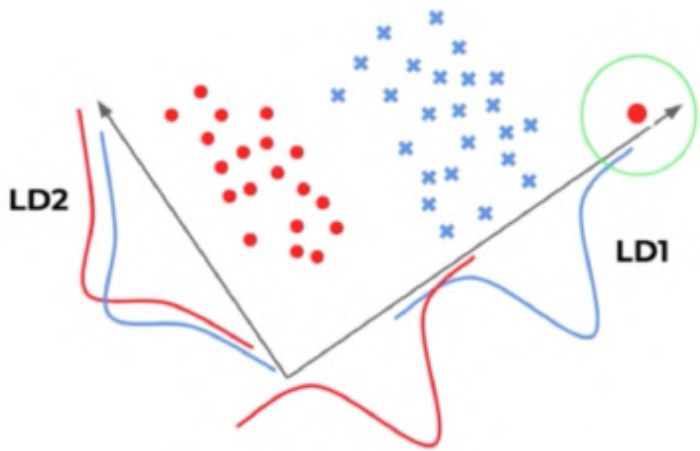


Why difference?

- When fitting an ARMA model, we want the data to be close to stationary
- Stationary = Does not change over time
- Stationarity is nice: mean, variance, autocorrelation, ... will be constant over time



- We want the data in our dataset all come from the same distribution
- What if our salary data is for software engineers
- But one data point comes from a marketer - this will bias the results!



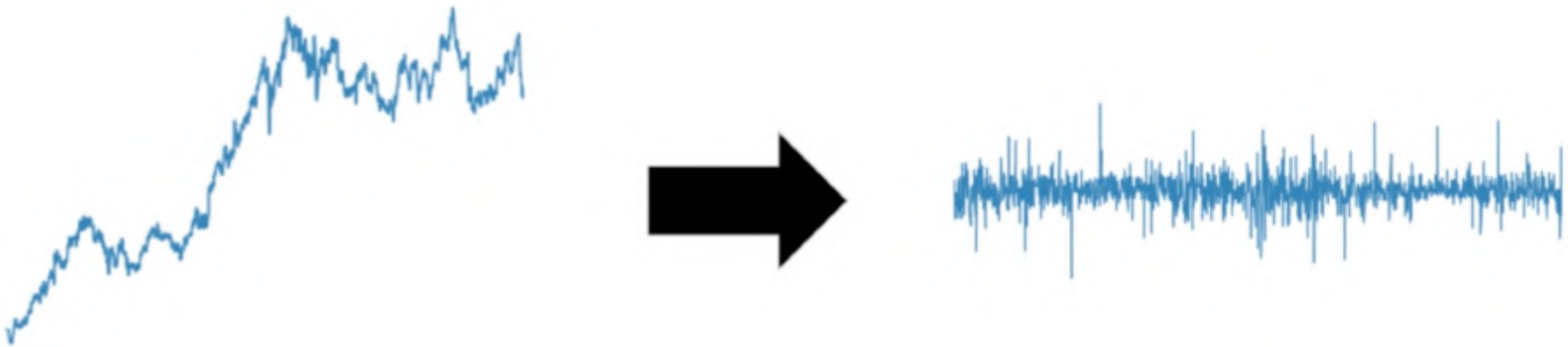
	Years of Experience	Age
Alice	10	35
Bob	5	30
Carol	2	25

Differencing and Stationarity

Differencing and Stationarity

- Differencing will often make the time series stationary!

- Sometimes, need to difference twice (but usually not more)



I(d) and ARIMA(p, d, q)

- An I(d) process is a process that is stationary after differencing d times

I(d) and ARIMA(p, d, q)

- An I(d) process is a process that is stationary after differencing d times
- We say it's integrated to order d

Special Cases

- ARIMA(p, 0, 0) is AR(p)

Special Cases

- ARIMA(p, 0, 0) is AR(p)
- It's also ARMA(p, 0)

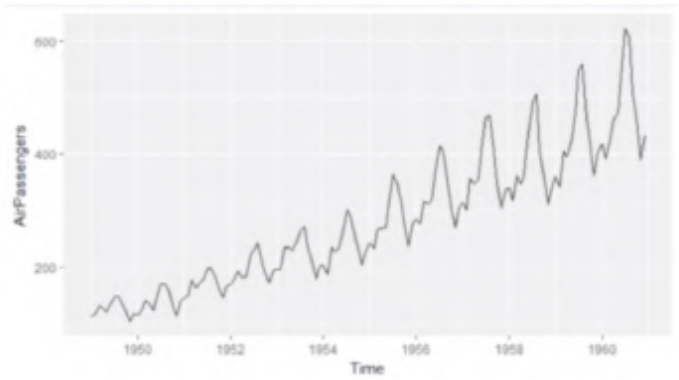
Special Cases

- ARIMA(p, 0, 0) is AR(p)
- It's also ARMA(p, 0)
- ARIMA(0, 0, q) is ARMA(0, q) and MA(q)

Hyper-Parameter Selection

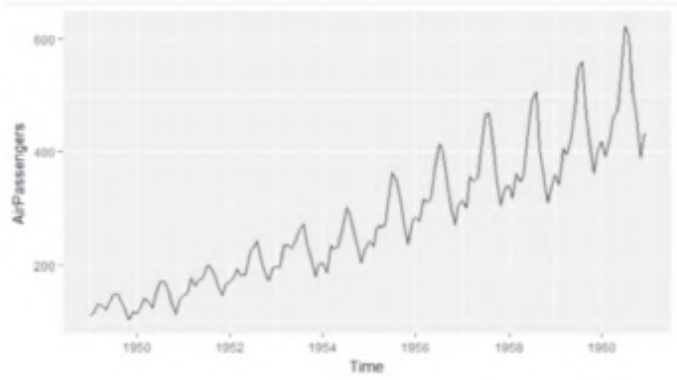
Stationarity and Hyperparameter Selection

- How to select p, d, q?
- In general, selecting hyperparameters isn't easy (e.g. deep learning)



Stationarity and Hyperparameter Selection

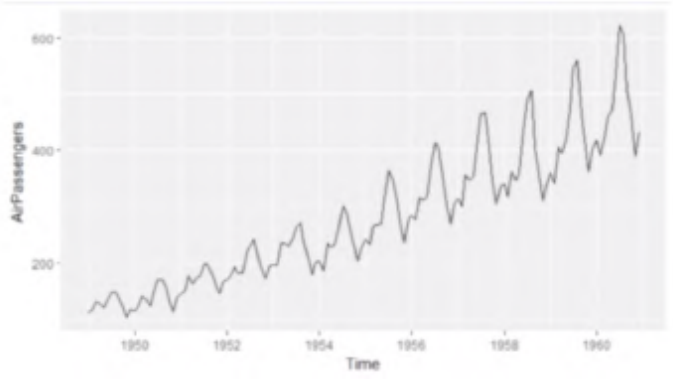
- How to select p, d, q?
- In general, selecting hyperparameters isn't easy (e.g. deep learning)
- There's no "formula", but rather trial and error



Stationarity and Hyperparameter Selection

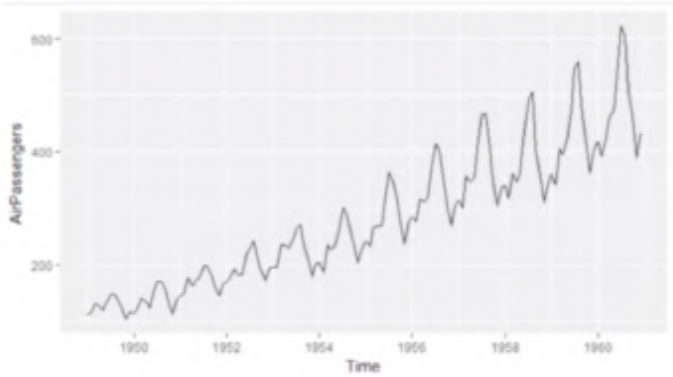
- How to select p, d, q?
- In general, selecting hyperparameters isn't easy (e.g. deep learning)
- There's no "formula", but rather trial and error

- But! For ARIMA, there is a "method" to select p, d, q

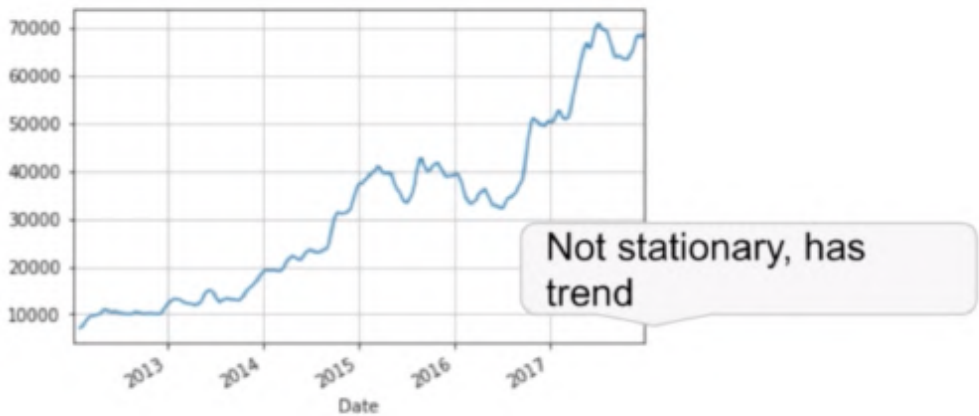


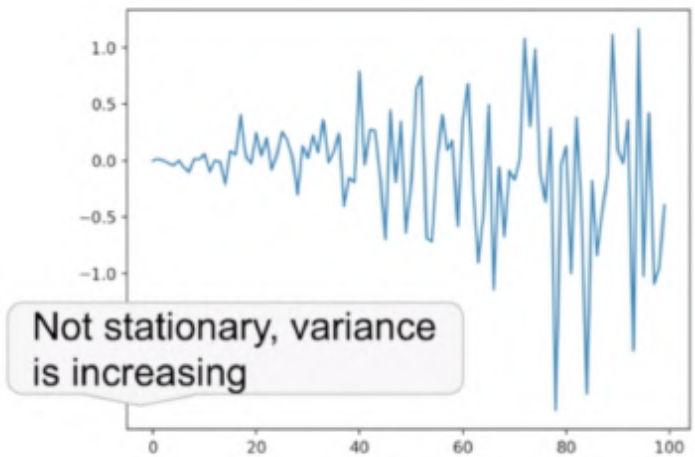
Stationarity and Hyperparameter Selection

- How to select p, d, q?
- In general, selecting hyperparameters isn't easy (e.g. deep learning)
- There's no "formula", but rather trial and error
- "Machine learning is experimentation, not philosophy"
- But! For ARIMA, there is a "method" to select p, d, q
- Stationarity will help us choose d



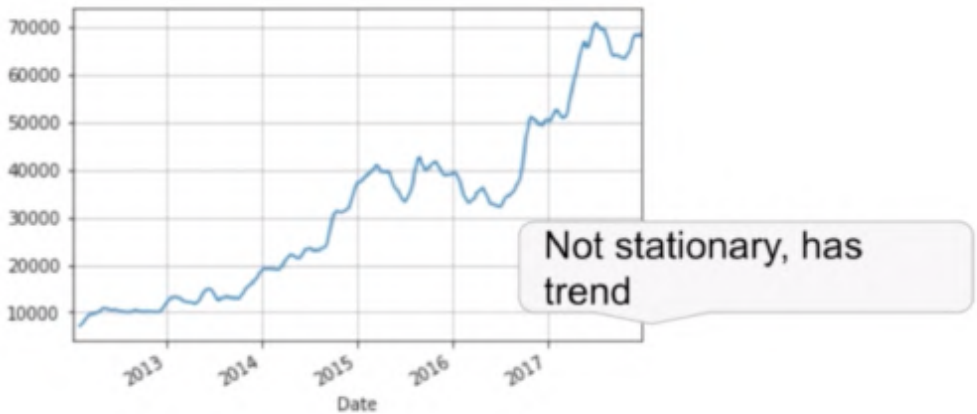
Practical explanation of stationarity

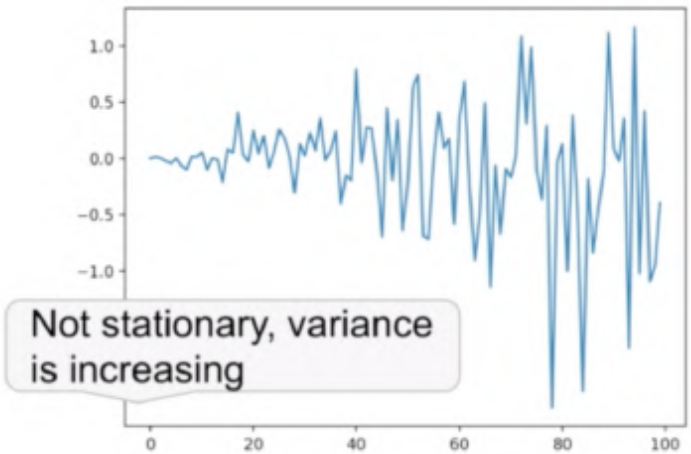




Practical explanation of stationarity

- Loosely, the distribution of the random variables in the time series does not change over time
- E.g. mean and variance will always be the same





Testing for stationarity

- We use the Augmented Dickey-Fuller Test (ADF Test)

Testing for stationarity

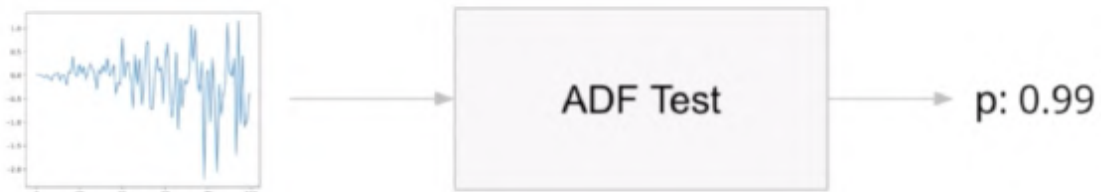
- We use the Augmented Dickey-Fuller Test (ADF Test)
- Think of it like an API:

Testing for stationarity

- We use the Augmented Dickey-Fuller Test (ADF Test)
- Think of it like an API:
  - Given: null hypothesis, alternative hypothesis

Testing for stationarity

- We use the Augmented Dickey-Fuller Test (ADF Test)
- Think of it like an API:
  - Given: null hypothesis, alternative hypothesis
  - Input: time series, Output: p-value



Testing for stationarity

- We use the Augmented Dickey-Fuller Test (ADF Test)
- Think of it like an API:
  - Given: null hypothesis, alternative hypothesis
  - Input: time series, Output: p-value
  - Action: accept or reject the null hypothesis

Testing for stationarity

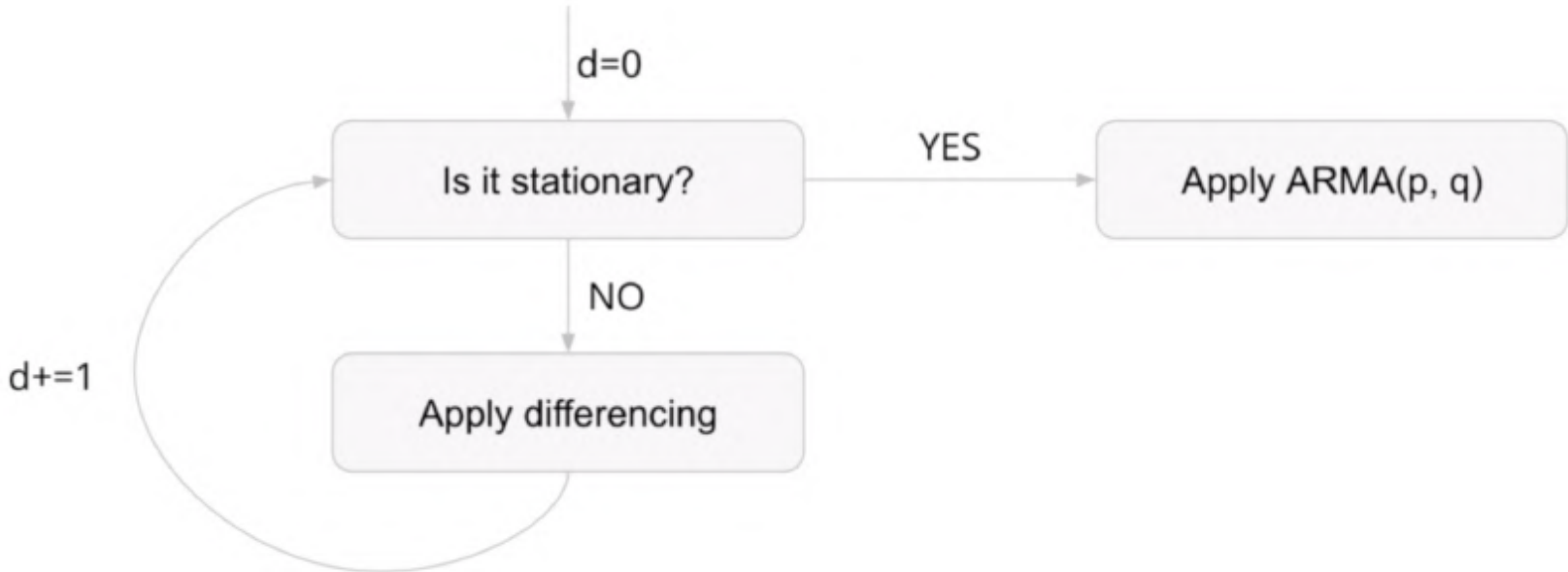
- We use the Augmented Dickey-Fuller Test (ADF Test)
- Think of it like an API:
  - Given: null hypothesis, alternative hypothesis
  - Input: time series, Output: p-value
  - Action: accept or reject the null hypothesis
- For ADF test:
  - Null: time series is non-stationary

Testing for stationarity

- We use the Augmented Dickey-Fuller Test (ADF Test)
- Think of it like an API:
  - Given: null hypothesis, alternative hypothesis
  - Input: time series, Output: p-value
  - Action: accept or reject the null hypothesis
- For ADF test:
  - Null: time series is non-stationary
  - Alternative: time series is stationary

How to use the ADF Test to select d

- Idea: keep differencing until the result is stationary (check using ADF)



How to choose q

- Let's continue our discussion of how to select ARIMA hyperparameters
- Interestingly, it is most convenient to look at d → q → p (opposite order we introduced AR, MA, and I components)

ACF - Autocorrelation Function

$$Autocovariance: \ cov(Y_{t_1}, Y_{t_2})$$

$$Autocorrelation: \ \frac{cov(Y_{t_1}, Y_{t_2})}{\sigma_Y(t_1)\sigma_Y(t_2)}$$

ACF - Autocorrelation Function

- Also known as correlogram
- Autocorrelation is to autocovariance as correlation is to covariance

$$Autocovariance: \ cov(Y_{t_1}, Y_{t_2})$$

$$Autocorrelation: \ \frac{cov(Y_{t_1}, Y_{t_2})}{\sigma_Y(t_1)\sigma_Y(t_2)}$$

ACF - Autocorrelation Function

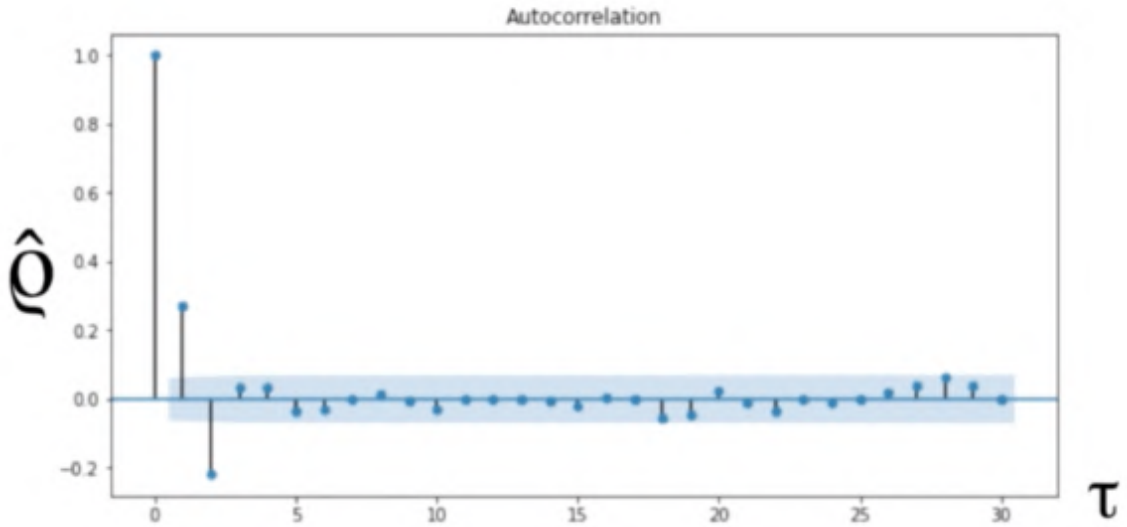
- Also known as correlogram
- Autocorrelation is to autocovariance as correlation is to covariance
- Auto = Self (both RVs come from the same time series)

$$Autocovariance: \ cov(Y_{t_1}, Y_{t_2})$$

$$Autocorrelation: \ \frac{cov(Y_{t_1}, Y_{t_2})}{\sigma_Y(t_1)\sigma_Y(t_2)}$$

Plotting the ACF

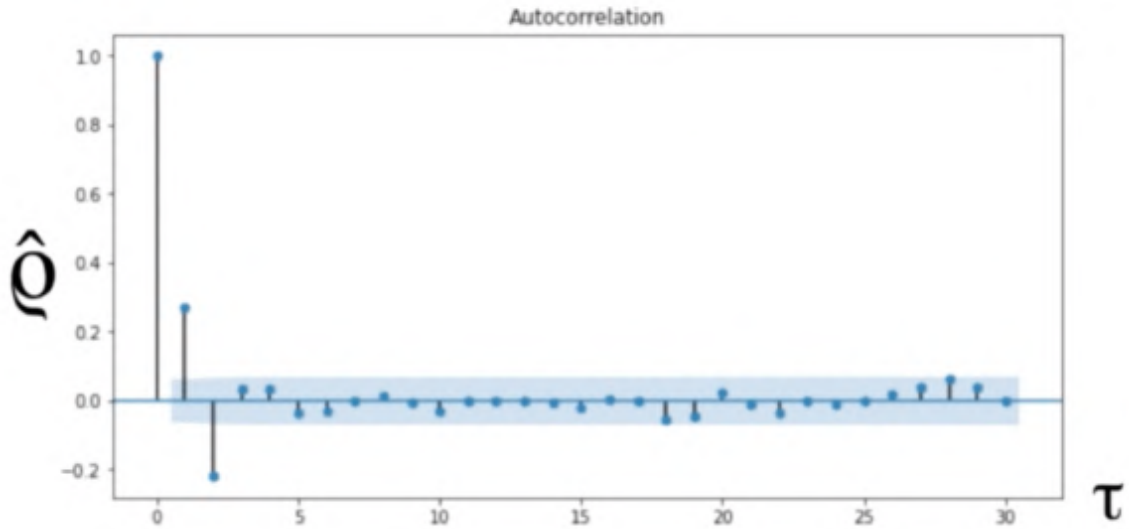
- Plot ACF value vs the lag (τ)
- In Python, see Scipy or statsmodels



Plotting the ACF

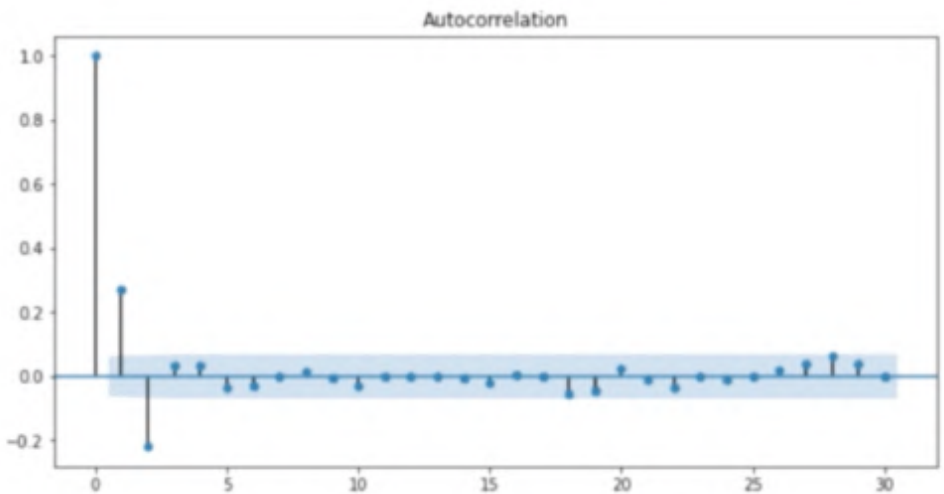
- Plot ACF value vs the lag (τ)
- In Python, see Scipy or statsmodels

At lag=0, we just get variance / variance = 1



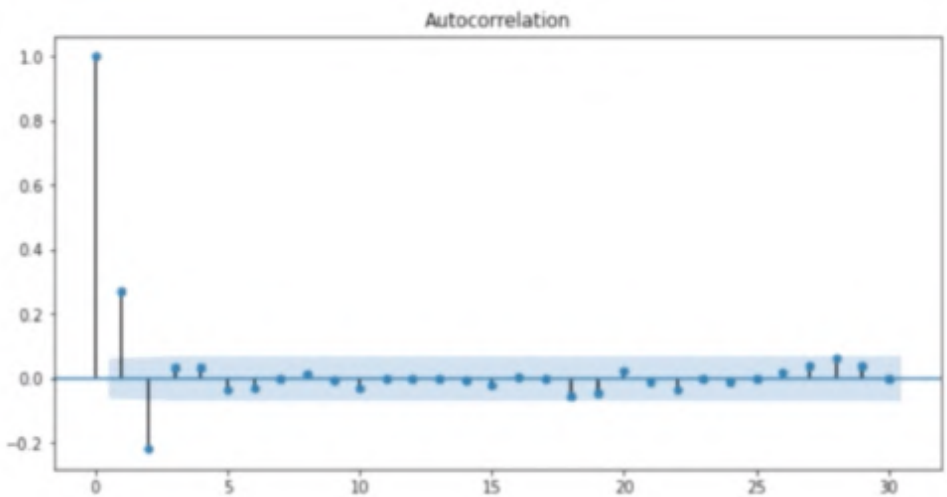
Confidence intervals in ACF plot

- If we see any lagged autocorrelations outside the confidence threshold, we will *reject* that they are equal to 0



How to determine q in MA(q)

- Assign q to be the maximum non-zero lag
- E.g. in below chart, q = 2
- Usually, the ACF for lags < q are also non-zero



PACF for AR(p) order selection

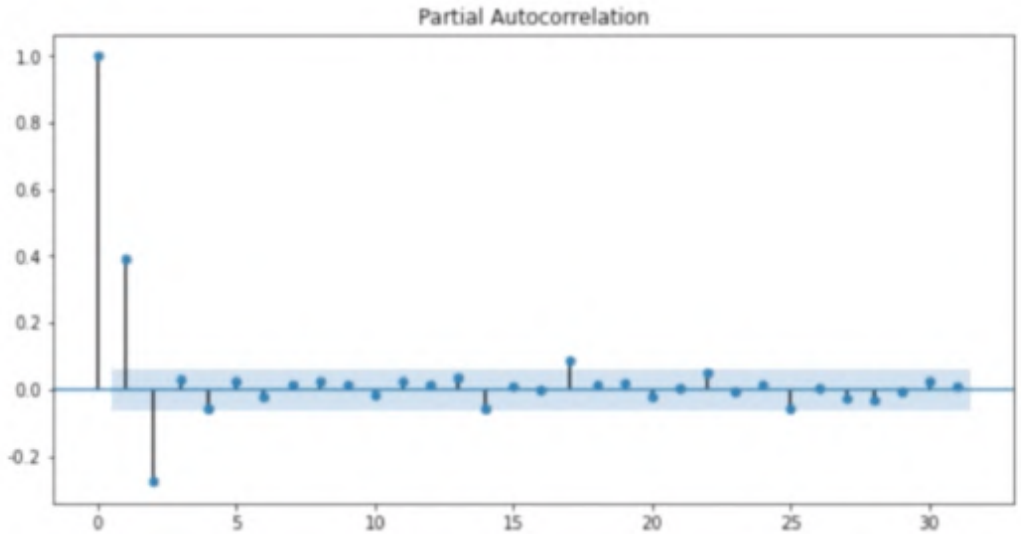
- PACF = Partial Autocorrelation Function

PACF for AR(p) order selection

- PACF = Partial Autocorrelation Function
- A bit more difficult to understand than ACF

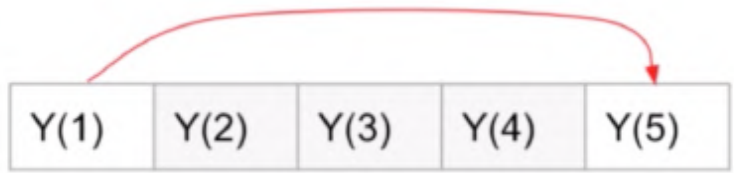
PACF for AR(p) order selection

- PACF = Partial Autocorrelation Function
- A bit more difficult to understand than ACF
- But its application to choosing p is the same



PACF for arbitrary τ

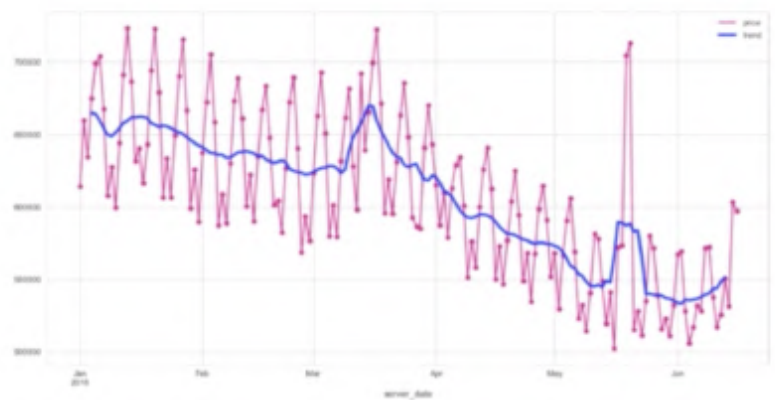
$$\hat{Y}_t = \beta_0' + \beta_1' Y_{t-1} + \beta_2' Y_{t-2} + \ldots \beta_\tau \quad' Y_{t-\tau}$$



SARIMA(p,d,q)x(P,D,Q)m

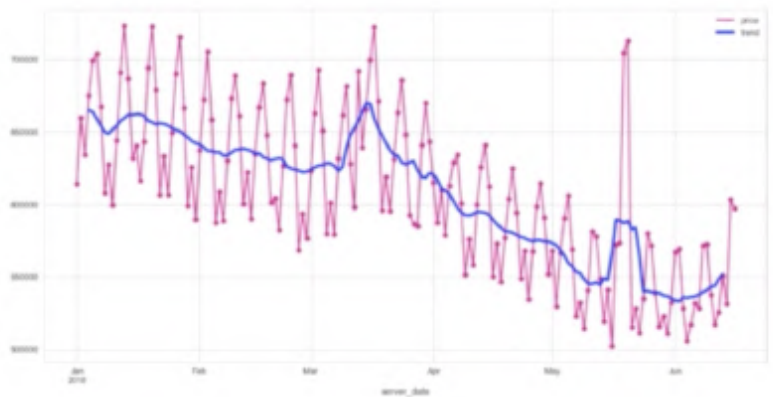
Seasonal ARIMA (SARIMA)

- E.g. weather (seasons) affect real estate purchases



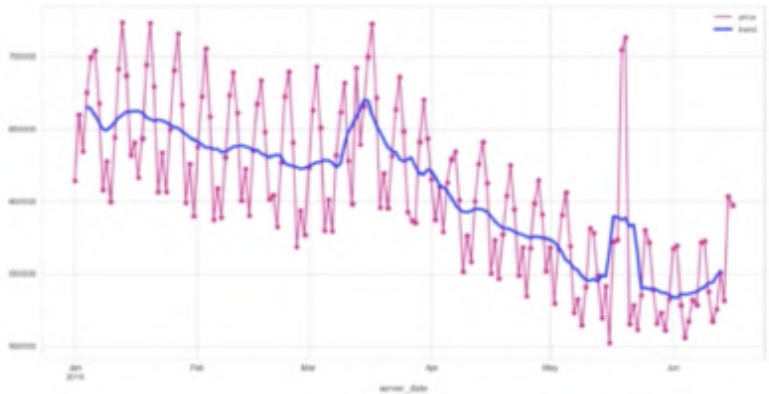
Seasonal ARIMA (SARIMA)

- E.g. weather (seasons) affect real estate purchases
- Notation: SARIMA{ (p, d, q) x $(P, D, Q)_m$ }



Seasonal ARIMA (SARIMA)

- E.g. weather (seasons) affect real estate purchases
- Notation: SARIMA{ (p, d, q) x $(P, D, Q)_m$ }
- If you write down the SARIMA model in "operator"-form, it is the multiplication of the non-seasonal and seasonal parts



Seasonal Differencing

If D=1:

$$\Delta_m Y_t = Y_t - Y_{t-m}$$

Seasonal Differencing

If D=1:

$$\Delta_m Y_t = Y_t - Y_{t-m}$$

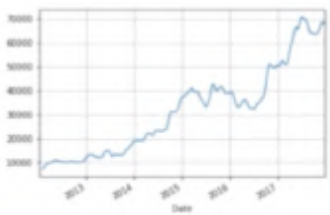
- E.g. if t = March, subtract the previous March

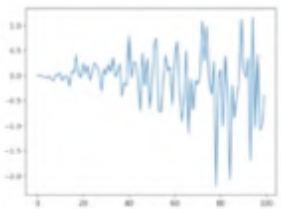
Seasonal Differencing

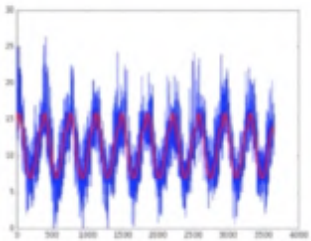
If D=1:

$$\Delta_m Y_t = Y_t - Y_{t-m}$$

- E.g. if t = March, subtract the previous March
- Add-on to our understanding of stationarity: it should not exhibit trend, change in variance, or seasonality



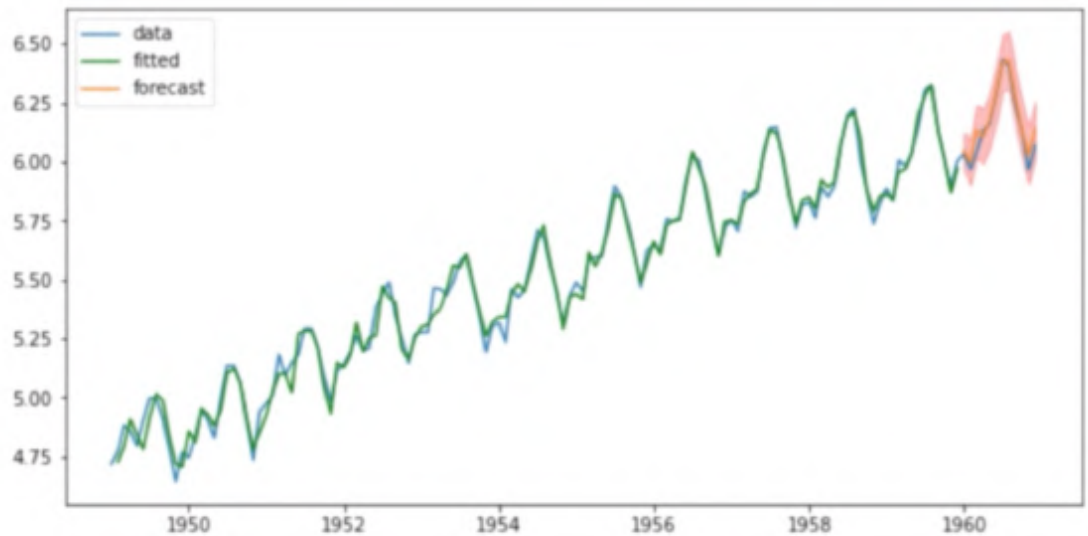




All of these are non-stationary

ARIMA without Seasonality

- It can still model seasonal time series, just not as nicely
- Fact: AR(2) can perfectly model a sine wave



Seasonal AR and MA

- Let's suppose Y(t) now represents the time series after seasonal differencing

$$AR\ part:\ Y_t = b + \varepsilon_t + \varphi_1 Y_{t-m} + \varphi_2 Y_{t-2m} + \ldots + \varphi_P Y_{t-Pm}$$

$$MA\ part:\ Y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-m} + \theta_2 \varepsilon_{t-2m} + \ldots + \theta_Q \varepsilon_{t-Qm}$$

Seasonal AR and MA

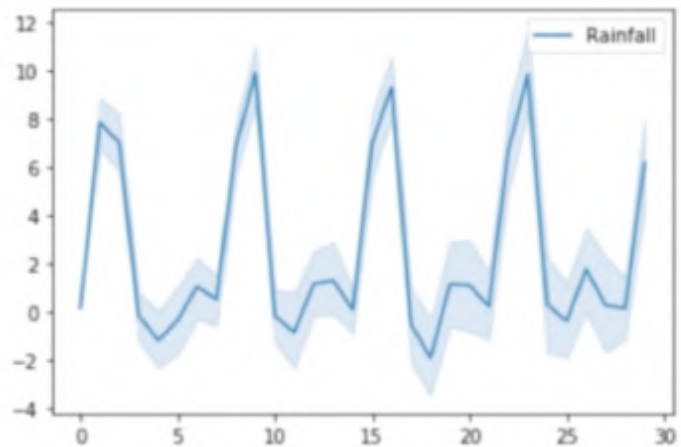
- Let's suppose Y(t) now represents the time series after seasonal differencing
- SARIMA is the seasonal AR component multiplied by the non-seasonal AR component, added to the season MA component multiplied by the non-seasonal MA component

$$AR\ part:\ Y_t = b + \varepsilon_t + \varphi_1 Y_{t-m} + \varphi_2 Y_{t-2m} + \ldots + \varphi_P Y_{t-Pm}$$

$$MA\ part:\ Y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-m} + \theta_2 \varepsilon_{t-2m} + \ldots + \theta_Q \varepsilon_{t-Qm}$$

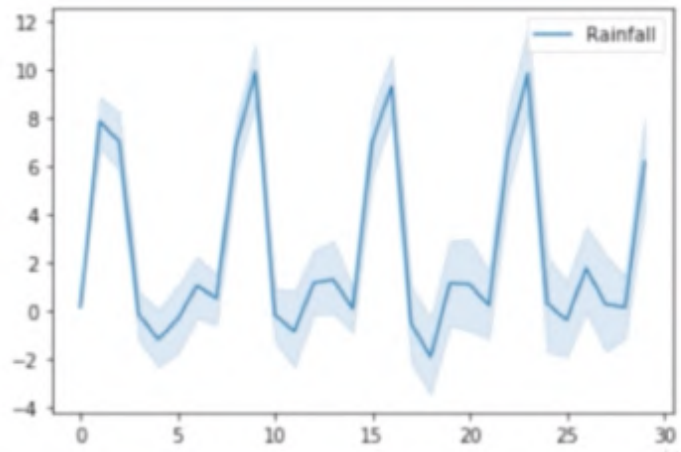
SARIMAX

- "X" refers to exogenous variables



SARIMAX

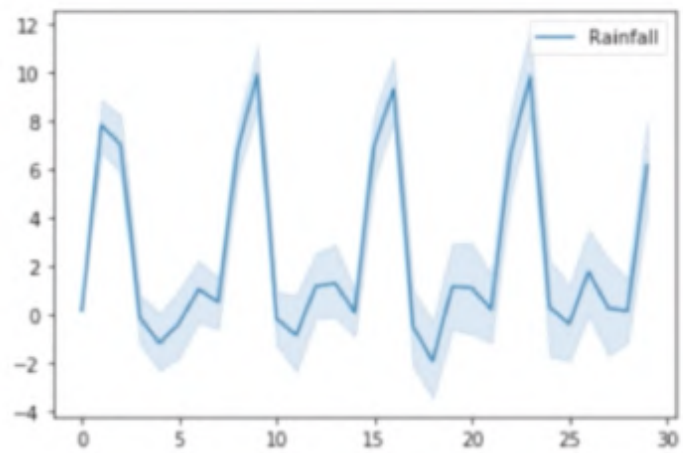
- "X" refers to exogenous variables
- Suppose you have a time series of length T



SARIMAX

- "X" refers to exogenous variables
- Suppose you have a time series of length T

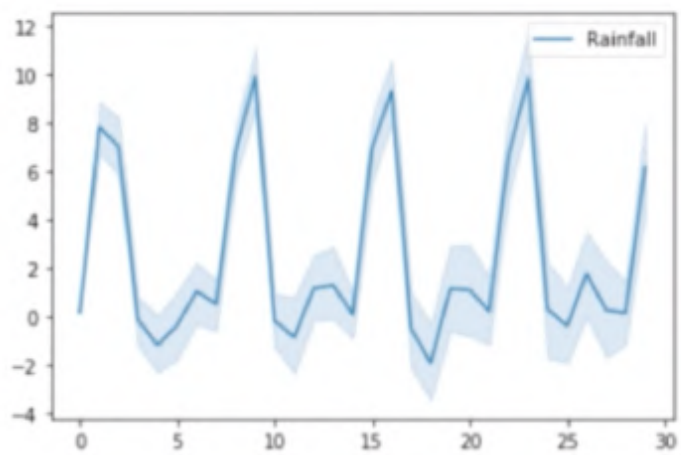
- You'll have an array of feature vectors (T x D)



SARIMAX

- "X" refers to exogenous variables
- Suppose you have a time series of length T

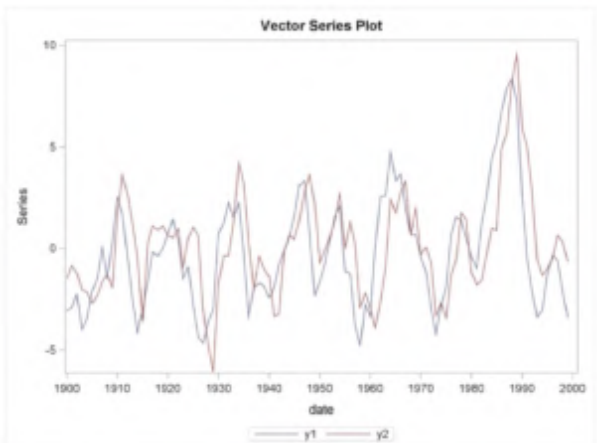
- You'll have an array of feature vectors (T x D)
- This can be passed into Auto ARIMA



VARMA model

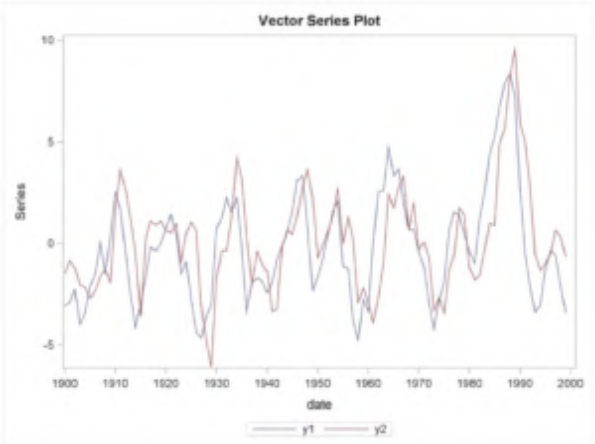
Vector Autoregressive and Moving Average Models

- Previously, we looked at only single time series' in isolation



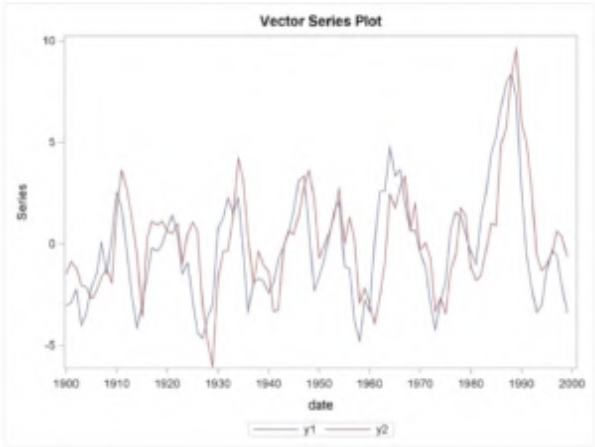
Vector Autoregressive and Moving Average Models

- Previously, we looked at only single time series' in isolation
- Assumption: the time series is predictable using only its past values



Vector Autoregressive and Moving Average Models

- Previously, we looked at only single time series' in isolation
- Assumption: the time series is predictable using only its past values
- In the real-world, could one time series affect another?



VARMA

$$\vec{y}_t = \vec{b} + \Phi_1 \vec{y}_{t-1} + \ldots + \Phi_p \vec{y}_{t-p} + \Theta_1 \vec{\varepsilon}_{t-1} + \ldots + \Theta_q \vec{\varepsilon}_{t-q} + \vec{\varepsilon}_t$$

VARMA

Previously scalars, now vectors!

$$\vec{y}_t = \vec{b} + \Phi_1 \vec{y}_{t-1} + \ldots + \Phi_p \vec{y}_{t-p} + \Theta_1 \vec{\varepsilon}_{t-1} + \ldots + \Theta_q \vec{\varepsilon}_{t-q} + \vec{\varepsilon}_t$$

VARMA

Previously scalars, now vectors!

Previously scalars, now matrices!

$$\vec{y}_t = \vec{b} + \Phi_1 \vec{y}_{t-1} + \ldots + \Phi_p \vec{y}_{t-p} + \Theta_1 \vec{\varepsilon}_{t-1} + \ldots + \Theta_q \vec{\varepsilon}_{t-q} + \vec{\varepsilon}_t$$

VARMA

ARMA for comparison

$$y_t = b + \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Previously scalars, now vectors!

Previously scalars, now matrices!

Capitalized

$$\vec{y}_t = \vec{b} + \Phi_1 \vec{y}_{t-1} + \ldots + \Phi_p \vec{y}_{t-p} + \Theta_1 \vec{\varepsilon}_{t-1} + \ldots + \Theta_q \vec{\varepsilon}_{t-q} + \vec{\varepsilon}_t$$

Normally I won't use arrows to denote vectors, it should be clear from context

VARMA

Called VAR(p) and VMA(q) as you'd expect

Vectors of size D

Still a linear model!

Pure autoregressive

$$\vec{y}_t = \vec{b} + \Phi_1 \vec{y}_{t-1} + \ldots + \Phi_p \vec{y}_{t-p} + \Theta_1 \vec{\varepsilon}_{t-1} + \ldots + \Theta_q \vec{\varepsilon}_{t-q} + \vec{\varepsilon}_t$$

Pure moving average

Matrices of size DxD

More Details

- Consider VAR(p) in 2 dimensions (let p=1 for simplicity)

$$\begin{bmatrix} y^{(1)}{}_t \\ y^{(2)}{}_t \end{bmatrix} = \begin{bmatrix} b^{(1)} \\ b^{(2)} \end{bmatrix} + \begin{bmatrix} \varphi_{11} & \varphi_{12} \\ \varphi_{21} & \varphi_{22} \end{bmatrix} \begin{bmatrix} y^{(1)}{}_{t-1} \\ y^{(2)}{}_{t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon^{(1)}{}_t \\ \varepsilon^{(2)}{}_t \end{bmatrix}$$

More Details

- Consider VAR(p) in 2 dimensions (let p=1 for simplicity)

$$\begin{bmatrix} y^{(1)}{}_t \\ y^{(2)}{}_t \end{bmatrix} = \begin{bmatrix} b^{(1)} \\ b^{(2)} \end{bmatrix} + \begin{bmatrix} \varphi_{11} & \varphi_{12} \\ \varphi_{21} & \varphi_{22} \end{bmatrix} \begin{bmatrix} y^{(1)}{}_{t-1} \\ y^{(2)}{}_{t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon^{(1)}{}_t \\ \varepsilon^{(2)}{}_t \end{bmatrix}$$

- Multiply it out and express both components in scalar form

$$y^{(1)}{}_t = b^{(1)} + \varphi_{11} y^{(1)}{}_{t-1} + \varphi_{12} y^{(2)}{}_{t-1} + \varepsilon^{(1)}{}_t$$

$$y^{(2)}{}_t = b^{(2)} + \varphi_{21} y^{(1)}{}_{t-1} + \varphi_{22} y^{(2)}{}_{t-1} + \varepsilon^{(2)}{}_t$$

More Details

- VAR(1)

$$y^{(1)}{}_t = b^{(1)} + \varphi_{11} y^{(1)}{}_{t-1} + \varphi_{12} y^{(2)}{}_{t-1} + \varepsilon^{(1)}{}_t$$
$$y^{(2)}{}_t = b^{(2)} + \varphi_{21} y^{(1)}{}_{t-1} + \varphi_{22} y^{(2)}{}_{t-1} + \varepsilon^{(2)}{}_t$$

- 2 separate AR(1)'s

$$y^{(1)}{}_t = b^{(1)} + \varphi_{11} y^{(1)}{}_{t-1} + \varepsilon^{(1)}{}_t$$
$$y^{(2)}{}_t = b^{(2)} + \varphi_{22} y^{(2)}{}_{t-1} + \varepsilon^{(2)}{}_t$$

How many parameters?

VAR for simplicity

- Scalar case: D(p + 1) parameters (D equations, p+1 each)

$$\hat{y}_t = b + \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p}$$

$$\ldots$$

$$\hat{y}_t = b + \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p}$$

How many parameters?

VAR for simplicity

- Scalar case: D(p + 1) parameters (D equations, p+1 each)

$$\hat{y}_t = b + \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p}$$

$$\ldots$$

$$\hat{y}_t = b + \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p}$$

- Vector case: $pD^2 + D$ parameters (p DxD matrices, and D-size bias term)

$$\hat{y}_t = b + \Phi_1 y_{t-1} + \ldots + \Phi_p y_{t-p}$$

Model Identifiability

- Fact: VARMA models are not unique
- Cannot uniquely determine p and q
- We'll see this warning in statsmodels

Sparse Identification and Estimation of Large-Scale Vector AutoRegressive Moving Averages

Ines Wilms[a]*, Sumanta Basu[b]*, Jacob Bien[c]†, and David S. Matteson[b]

[a] *Department of Quantitative Economics, Maastricht University, Maastricht, The Netherlands*

[b] *Department of Statistics and Data Science, Cornell University, Ithaca, NY, USA*

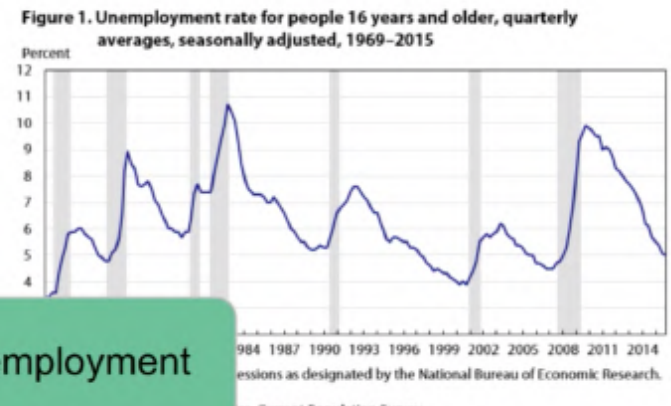
[c] *Data Sciences and Operations, University of Southern California, Los Angeles, CA, USA*

Abstract. The Vector AutoRegressive Moving Average (VARMA) model is fundamental to the theory of multivariate time series; however, identifiability issues have led practitioners to abandon it in favor of the simpler but more restrictive Vector AutoRegressive (VAR) model.

VARIMA?

- We only want to use VARMA / ARMA if the time series is stationary

GDP

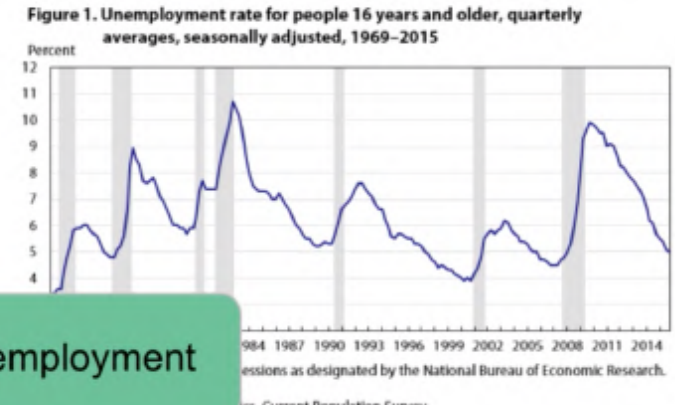


Unemployment

VARIMA?

- We only want to use VARMA / ARMA if the time series is stationary
- Can find some packages / discussion for VARIMA, but not in statsmodels

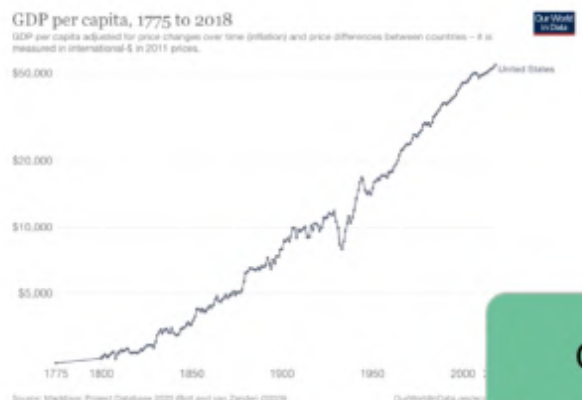
GDP



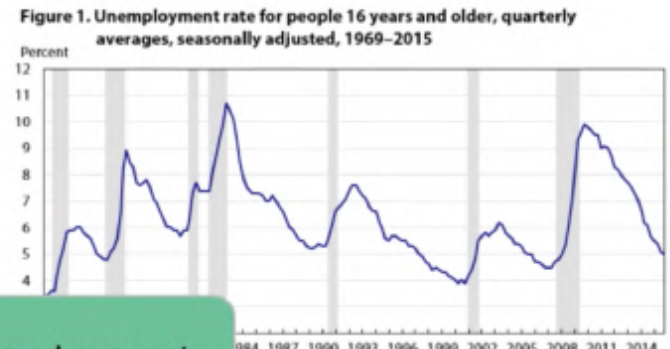
Unemployment

VARIMA?

- We only want to use VARMA / ARMA if the time series is stationary
- Can find some packages / discussion for VARIMA, but not in statsmodels
- Each component of your time series may need to be differenced a different number of times than the others! (i.e. there isn't one "d" value)



GDP



Unemployment

VARIMA?

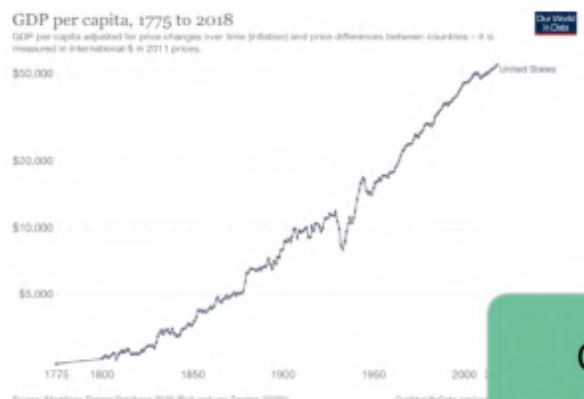
- We only want to use VARMA / ARMA if the time series is stationary
- Can find some packages / discussion for VARIMA, but not in statsmodels
- Each component of your time series may need to be differenced a different number of times than the others! (i.e. there isn't one "d" value)

GDP

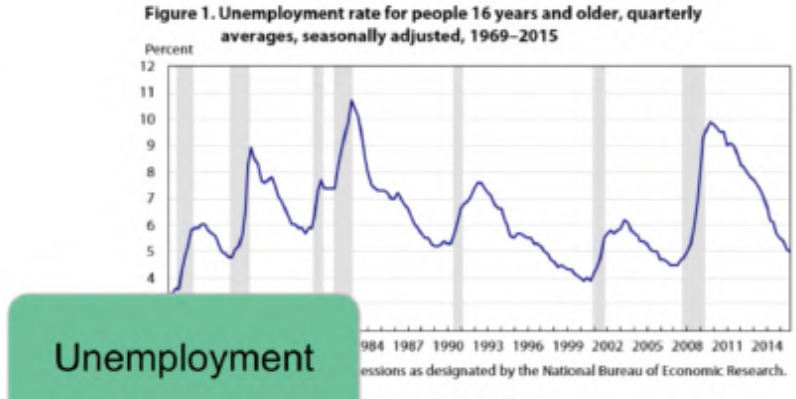
Unemployment

VARIMA?

- We only want to use VARMA / ARMA if the time series is stationary
- Can find some packages / discussion for VARIMA, but not in statsmodels
- Each component of your time series may need to be differenced a different number of times than the others! (i.e. there isn't one "d" value)



GDP



Unemployment

Summary

- VARMA(p, q) is just an extension of ARMA(p, q) to vectors

Summary

- VARMA(p, q) is just an extension of ARMA(p, q) to vectors
- Still linear; each component depends on lags of every other component

Summary

- VARMA(p, q) is just an extension of ARMA(p, q) to vectors
- Still linear; each component depends on lags of every other component
- VARMA is not identifiable

Summary

- VARMA(p, q) is just an extension of ARMA(p, q) to vectors
- Still linear; each component depends on lags of every other component
- VARMA is not identifiable
- VARMA takes longer to train

Summary

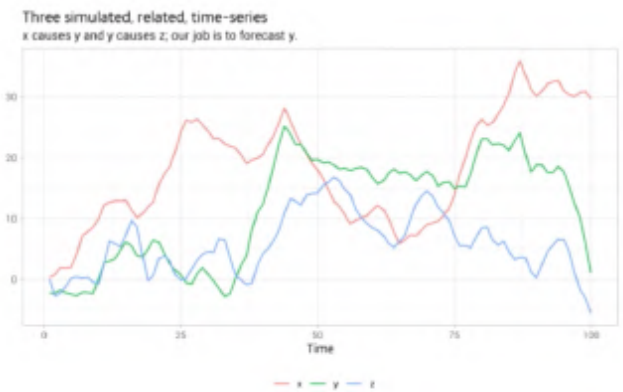
- VARMA(p, q) is just an extension of ARMA(p, q) to vectors
- Still linear; each component depends on lags of every other component
- VARMA is not identifiable
- VARMA takes longer to train
- Number of parameters grows quadratically with number of components

Summary

- VARMA(p, q) is just an extension of ARMA(p, q) to vectors
- Still linear; each component depends on lags of every other component
- VARMA is not identifiable
- VARMA takes longer to train
- Number of parameters grows quadratically with number of components
- May lead to overfitting (too many parameters)

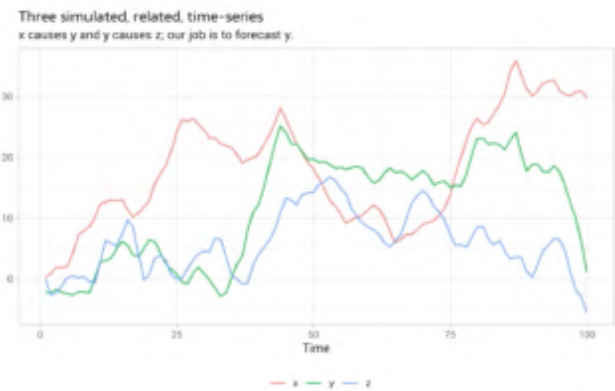
VARMA Summary

- In statsmodels, differencing must be done manually (each component of the time series could have a different integration order)



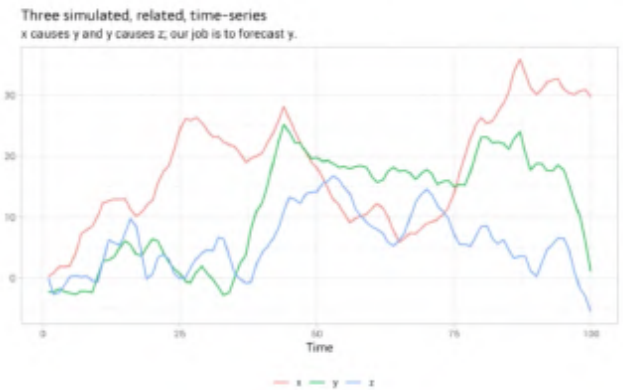
VARMA Summary

- In statsmodels, differencing must be done manually (each component of the time series could have a different integration order)
- This is not a “scientific” observation - it’s just how statsmodels works
- Other languages / libraries look like they have VARIMA capabilities



VARMA Summary

- In statsmodels, differencing must be done manually (each component of the time series could have a different integration order)
- This is not a “scientific” observation - it’s just how statsmodels works
- Other languages / libraries look like they have VARIMA capabilities
- If you difference, then you must undo differencing when forecasting!



Transformations of Time Series

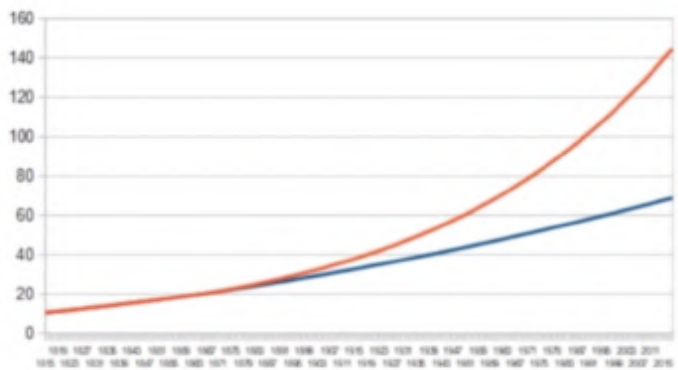
Time Series Transformations

- If you're familiar with ML, then you know why it's useful to transform your data (e.g. standardization, min-max scaling)
- For time series, we'll discuss:

1) Power transform
2) Log transform
3) Box-Cox transform

Power Transform

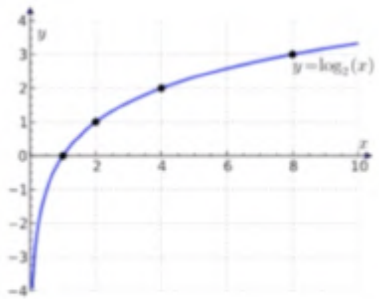
- Raise all data points to a power (e.g. $\gamma = ½$)



$$y'(t) = y(t)^{\gamma}$$

Log Transform

- "Squashes" the data (similar to power transform, but power transform can do the opposite too)

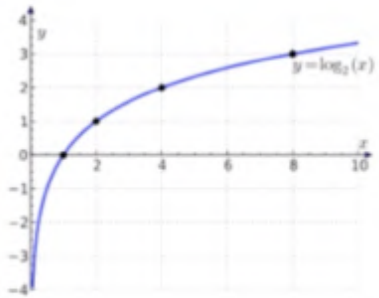


Note: I often use log() and ln() interchangeably

$$y'(t) = \log y(t) \; or \; \log (y(t) + 1)$$

Log Transform

- "Squashes" the data (similar to power transform, but power transform can do the opposite too)
- I often use the log transform by default
- Common application: finance
  - Prices are modeled as a log-normal distribution



Note: I often use log() and ln() interchangeably

$$y'(t) = \log y(t) \text{ or } \log(y(t) + 1)$$

Box-Cox Transform

- Unifies the power and log transforms
- λ is chosen automatically by the boxcox() function in Scipy
  - See documentation and extra_reading.txt for details
  - "Estimating Box-Cox power transformation parameter via goodness of fit tests"

$$y'(t) = \frac{y(t)^{\lambda} - 1}{\lambda} \quad if\ \lambda \neq 0$$

$$y'(t) = \log y(t) \quad if\ \lambda = 0$$

Since:

$$\lim_{\lambda \to 0} \frac{x^{\lambda}-1}{\lambda} = \ln x$$

Let's get Started!

Access Google Colaboratory through your Gmail account